Structural Engineering and Mechanics, *Vol. 52, No. 6 (2014) 1121-1134*
DOI: http://dx.doi.org/10.12989/sem.2014.52.6.1121

An efficient method to structural static reanalysis with deleting support constraints

Haifeng Liu[*1] and Shigang Yue[2]

[1]*Department of Mathematics, School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, P.R. China*
[2]*School of Computer Science, University of Lincoln, Brayford Pool Campus, Lincoln LN6 7TS, UK*

(Received October 1, 2013, Revised May 20, 2014, Accepted June 12, 2014)

Abstract. Structural design is usually an optimization process. Numerous parameters such as the member shapes and sizes, the elasticity modulus of material, the locations of nodes and the support constraints can be selected as design variables. These variables are progressively revised in order to obtain a satisfactory structure. Each modification requires a fresh analysis for the displacements and stresses, and reanalysis can be employed to reduce the computational cost. This paper is focused on static reanalysis problem with modification of deleting some supports. An efficient reanalysis method is proposed. The method makes full use of the initial information and preserves the ease of implementation. Numerical examples show that the calculated results of the proposed method are the identical as those of the direct analysis, while the computational time is remarkably reduced.

Keywords: structural reanalysis; modification of supports; Cholesky factorization; stiffness matrix; computational cost

1. Introduction

Many large complex structures require to be designed in some fields such as civil engineering, petrochemical engineering and aerospace engineering. Structural analysis is indispensable during the process of design. Generally speaking, the design may be modified many times until a satisfactory structure is obtained. Each modification involves a fresh analysis. These repeated analyses cost much computational time and lead to an unacceptable numerical burden (Chen and Huang 2013). In order to improve the case, structural static reanalysis problem has been proposed. The objective of structural static reanalysis is to calculate the response of the modified structure by making full use of the information from the initial analysis so that the computational cost can be greatly reduced (Abu Kassim and Topping 1987). Static reanalysis is significant for designing large structures, especially for the case in which only a small part of the structure is progressively modified (Terdalkar and Rencis 2006).

Some static reanalysis methods have already been proposed. Generally speaking, these methods can be divided into two classes: direct reanalysis methods and approximate reanalysis methods.

∗Corresponding author, Ph.D., E-mail: lhflkcc@mail.xjtu.edu.cn

http://www.techno-press.org/?journal=sem&subpage=8 ISSN: 1225-4568 (Print), 1598-6217 (Online)

Direct methods provide exact closed-form solutions of the response. The computational costs of these methods depend on the number of the changed elements, and are unrelated to the extent of the changes. For this reason, direct methods are suitable for the modifications where the changes in design variables are large in magnitude, yet only affect a small part of the structure. Most of these methods utilize the famous Sherman-Morrison-Woodbury formulae, and several improvements and variations have been proposed (Akgün *et al.* 2001, Pais *et al.* 2012).

Approximate reanalysis methods present the approximate solutions of the response. The accuracy of the approximate solution and the convergent speed are two key issues for these methods. Approximate methods can be broadly classified into four categories: local approximations, global approximations, combined approximations (CA), and preconditioned conjugate gradient (PCG) approximations. For the details of these approximate methods, we refer the readers to Kirsch (2008), Li and Wu (2007). Among the above four approximate methods, PCG approximations are the most efficient. Recently, a PCG approximation for unchanged number of degrees of freedom (DOFs) was proposed (Liu *et al.* 2012a). The method utilizes the algorithm of updating Cholesky factorization to construct a new preconditioner. It is especially efficient in dealing with the cases in which small parts of the structure are heavily modified while major parts are slightly modified. More importantly, an exact solution can be given by using the procedure of constructing the preconditioner when the number of changed elements is small.

The above mentioned reanalysis methods can deal with various structural modifications such as cross-sectional modifications, material modifications, geometrical modifications and layout modifications. However, structural supports as the design variables have been brought into wide use (Takezawa *et al.* 2006, Tanskanen 2006, Zhu and Zhang 2010), especially in the fields of building construction, aircraft structures and printed circuit boards (Wang *et al.* 2004). In addition, elastic contact problems, such as normal, tangential, and rolling contacts, can be transformed into multiple point constraints for nodal displacements in the finite elements analysis method (Liu *et al.* 2010). The reanalysis methods for such modifications are relatively few. The supporting modifications include the variations of the location, the number, and the type of the structural supports (Olhoff and Taylor 1983). A small change in supports has great effect on structural performance, especially on the nodal displacements and the natural frequency. Meanwhile, these modifications often result in the variations of the number of DOFs. Therefore, the reanalysis for the support modifications is a challenging problem in the field of structural reanalysis. Liu *et al.* (2012b) studied the case of adding some supports whose orientations are the same as that of some axes of the global coordinate system. The method requires to solve several linear system with the same coefficient matrix, i.e., the initial stiffness matrix, then a linear system with lower order is involved for calculating the response of the modified structure. The exact solution is provided. Thus, the method belongs to the direct reanalysis method.

This paper is a follow up to Liu *et al.* (2012b). Static reanalysis problem with modification of deleting some support constraints whose orientations are the same as that of some axes of global coordinate system is studied. The initial information has been fully utilized, a property of Cholesky factorization is studied and employed, finally a new method is proposed. The paper is organized as follows. The Cholesky factorization is reviewed, and its property related to our problem is studied in Section 2. The mathematical formulations of the considered problem are given in Section 3. Section 4 contains our method for dealing with the problem. Numerical examples are used to validate the effectiveness of the proposed method in Section 5, and finally some conclusions are drawn in Section 6.

2. Cholesky factorization and its property related to our problem

Cholesky factorization is a fundamental factorization for symmetric and positive definite (SPD) matrix, it is mainly used to solve SPD linear system. Any SPD matrix $\mathbf{A}\in R^{n\times n}$ can be factored in the form

$$\mathbf{A} = \mathbf{G}\mathbf{G}^{\mathrm{T}} \tag{1}$$

where $\mathbf{G}\in R^{n\times n}$ is a lower triangular matrix with positive entries on its diagonal, and $\mathbf{G}^{\mathrm{T}}$ denotes the transpose of **G**. This factorization is called the Cholesky factorization of **A**, and **G** is named as the Cholesky triangle. Given a SPD matrix **A**, its Cholesky factorization is unique (Golub and Van Loan 1996), i.e., if a lower triangular matrix $\mathbf{G}_1$ satisfies the following equation

$$\mathbf{A} = \mathbf{G}_1\mathbf{G}_1^{\mathrm{T}} \tag{2}$$

then we have $\mathbf{G}_1=\mathbf{G}$. Once the Cholesky factorization $\mathbf{A}=\mathbf{G}\mathbf{G}^{\mathrm{T}}$ has been calculated, the linear system $\mathbf{Ax}=\mathbf{b}$ can be solved via the two triangular systems $\mathbf{Gy}=\mathbf{b}$ and $\mathbf{G}^{\mathrm{T}}\mathbf{x}=\mathbf{y}$ with little cost. The algorithm for computing the Cholesky factorization can be derived by manipulating the equation $\mathbf{A}=\mathbf{G}\mathbf{G}^{\mathrm{T}}$ (Golub and Van Loan 1996).

Algorithm 1 (Cholesky factorization)

$g_{11} = \sqrt{a_{11}}$
For $i = 2, \cdots, n$
  for $j = 1, \cdots, i-1$
    for $k = 1, \cdots, j-1$
      $a_{ij} = a_{ij} - g_{ik}g_{jk}$
    end
    $g_{ij} = a_{ij} / g_{jj}$
  end
  for $k = 1, \cdots, i-1$
    $a_{ii} = a_{ii} - g_{ik}^2$
  end
  $g_{ii} = \sqrt{a_{ii}}$
End

This algorithm requires $(1/3)n^3$ floating point operations (flops) and is numerical stable. When the bandwidth of **A** is taken into account, the algorithm can be slightly revised and the computational cost is reduced, see Golub and Van Loan (1996). Cholesky factorization can be regarded as a variant of Gaussian elimination that operates on both the left and the right of the matrix at once, preserving and exploiting symmetry. For the further research on Cholesky factorization, see Davis (2006).

In order to deal with our problem, the following property of Cholesky factorization is introduced.

Property 1 Suppose $\mathbf{A}\in R^{n\times n}$ is SPD, and its Cholesky factorization is $\mathbf{A}=\mathbf{G}\mathbf{G}^{\mathrm{T}}$. Assume

$\mathbf{B}=\begin{bmatrix}\mathbf{A} & \mathbf{B}_{12}\\ \mathbf{B}_{12}^{\mathrm{T}} & \mathbf{B}_{22}\end{bmatrix}\in R^{(n+k)\times(n+k)}$ is also SPD, where $\mathbf{B}_{12}\in R^{n\times k}$, $\mathbf{B}_{22}\in R^{k\times k}$. Then its Cholesky factorization has the following form $\mathbf{B}=\begin{bmatrix}\mathbf{G} & \mathbf{0}\\ \mathbf{G}_{21} & \mathbf{G}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{G}^{\mathrm{T}} & \mathbf{G}_{21}^{\mathrm{T}}\\ \mathbf{0} & \mathbf{G}_{22}^{\mathrm{T}}\end{bmatrix}$, where $\mathbf{0}$ represents the corresponding zero matrix.

Proof Let the Cholesky factorization of **B** be partitioned in the form $\mathbf{B}=\begin{bmatrix}\mathbf{A} & \mathbf{B}_{12}\\ \mathbf{B}_{12}^{\mathrm{T}} & \mathbf{B}_{22}\end{bmatrix}=\begin{bmatrix}\mathbf{G}_{11} & \mathbf{0}\\ \mathbf{G}_{21} & \mathbf{G}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{G}_{11}^{\mathrm{T}} & \mathbf{G}_{21}^{\mathrm{T}}\\ \mathbf{0} & \mathbf{G}_{22}^{\mathrm{T}}\end{bmatrix}$, then $\mathbf{A}=\mathbf{G}_{11}\mathbf{G}_{11}^{\mathrm{T}}$ can be obtained by utilizing the multiplication of block matrix. From the uniqueness of Cholesky factorization, we have $\mathbf{G}_{11}=\mathbf{G}$, i.e., the Cholesky factorization of **B** has the form $\mathbf{B}=\begin{bmatrix}\mathbf{G} & \mathbf{0}\\ \mathbf{G}_{21} & \mathbf{G}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{G}^{\mathrm{T}} & \mathbf{G}_{21}^{\mathrm{T}}\\ \mathbf{0} & \mathbf{G}_{22}^{\mathrm{T}}\end{bmatrix}$.

An important role of the above property is that when the Cholesky factorization of **A** has already been known, it can be used to calculate the factorization of **B** instead of factoring **B** directly. It will be seen in the following that this property is especially valuable for our problem.

3. Mathematical formulations of the considered problem

The static reanalysis problem for deleting some support constraints whose orientations are the same as that of some axes of global coordinate system can be stated as follows.

Given an initial design, the corresponding stiffness matrix is $\mathbf{K}_0\in R^{m\times m}$. The displacement vector $\mathbf{y}_0$ can be achieved by solving the equilibrium equations

$$\mathbf{K}_0\mathbf{y}_0=\mathbf{r}_0 \tag{3}$$

where $\mathbf{r}_0\in R^m$ denotes the load vector. The stiffness matrix $\mathbf{K}_0$ is SPD. From the initial analysis, its Cholesky factorization has already been known

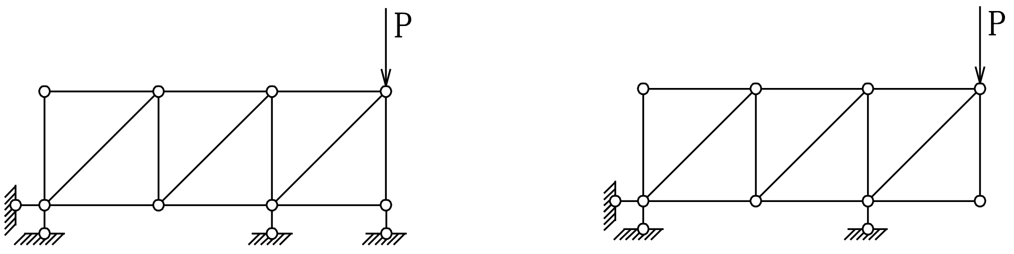
$$\mathbf{K}_0=\mathbf{L}_0\mathbf{L}_0^{\mathrm{T}} \tag{4}$$

where $\mathbf{L}_0$ is the Cholesky triangle of $\mathbf{K}_0$, and $\mathbf{L}_0^{\mathrm{T}}$ stands for the transpose of $\mathbf{L}_0$. Assume some supports are deleted, the number is *s*, and the orientations of these supports are the same as that of some axes of global coordinate system. The truss structure in Fig. 1 is employed to illustrate the type of modification that our proposed method can deal with. Suppose the initial structure is Fig. 1(a), the modification of deleting support constraint like Fig. 1(b) is studied, while modification of deleting skew support constraint is not considered in this paper.

Compared with the number of the initial DOFs, the number of deleted support constraints is usually very small, i.e., $s<<m$. The modified equilibrium equation is

$$\mathbf{K}\mathbf{x}=\mathbf{r} \tag{5}$$

where $\mathbf{K}\in R^{(m+s)\times(m+s)}$ is the modified stiffness matrix and is also SPD, $\mathbf{x}\in R^{m+s}$ represents the displacements vector and $\mathbf{r}\in R^{m+s}$ denotes the modified load vector. The relationship between K_0



(a) (b)

Fig. 1 Truss structure. (a) initial design, (b) modified design with deleting a support constraint along vertical axis of the global coordinate system

and **K** is given in the following. Assume

$$
\mathbf{K}_0 = \begin{bmatrix}
k_{11} & \cdots & k_{1i_1-1} & k_{1i_1+1} & \cdots & k_{1i_s-1} & k_{1i_s+1} & \cdots & k_{1m+s} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
k_{i_1-11} & \cdots & k_{i_1-1i_1-1} & k_{i_1-1i_1+1} & \cdots & k_{i_1-1i_s-1} & k_{i_1-1i_s+1} & \cdots & k_{i_1-1m+s} \\
k_{i_1+11} & \cdots & k_{i_1+1i_1-1} & k_{i_1+1i_1+1} & \cdots & k_{i_1+1i_s-1} & k_{i_1+1i_s+1} & \cdots & k_{i_1+1m+s} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
k_{i_s-11} & \cdots & k_{i_s-1i_1-1} & k_{i_s-1i_1+1} & \cdots & k_{i_s-1i_s-1} & k_{i_s-1i_s+1} & \cdots & k_{i_s-1m+s} \\
k_{i_s+11} & \cdots & k_{i_s+1i_1-1} & k_{i_s+1i_1+1} & \cdots & k_{i_s+1i_s-1} & k_{i_s+1i_s+1} & \cdots & k_{i_s+1m+s} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
k_{m+s1} & \cdots & k_{m+s\,i_1-1} & k_{m+s\,i_1+1} & \cdots & k_{m+s\,i_s-1} & k_{m+s\,i_s+1} & \cdots & k_{m+s\,m+s}
\end{bmatrix}_{m\times m} \tag{6}
$$

K can be obtained by inserting some rows and columns into $\mathbf{K}_0$ symmetrically, i.e.

$$
\mathbf{K} = \begin{bmatrix}
k_{11} & \cdots & k_{1i_1-1} & a_{1i_1} & k_{1i_1+1} & \cdots & k_{1i_s-1} & a_{1i_s} & k_{1i_s+1} & \cdots & k_{1m+s} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
k_{i_1-11} & \cdots & k_{i_1-1i_1-1} & a_{i_1-1i_1} & k_{i_1-1i_1+1} & \cdots & k_{i_1-1i_s-1} & a_{i_1-1i_s} & k_{i_1-1i_s+1} & \cdots & k_{i_1-1m+s} \\
a_{i_1 1} & \cdots & a_{i_1 i_1-1} & a_{i_1 i_1} & a_{i_1 i_1+1} & \cdots & a_{i_1 i_s-1} & a_{i_1 i_s} & a_{i_1 i_s+1} & \cdots & a_{i_1 m+s} \\
k_{i_1+11} & \cdots & k_{i_1+1i_1-1} & a_{i_1+1i_1} & k_{i_1+1i_1+1} & \cdots & k_{i_1+1i_s-1} & a_{i_1+1i_s} & k_{i_1+1i_s+1} & \cdots & k_{i_1+1m+s} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
k_{i_s-11} & \cdots & k_{i_s-1i_1-1} & a_{i_s-1i_1} & k_{i_s-1i_1+1} & \cdots & k_{i_s-1i_s-1} & a_{i_s-1i_s} & k_{i_s-1i_s+1} & \cdots & k_{i_s-1m+s} \\
a_{i_s 1} & \cdots & a_{i_s i_1-1} & a_{i_s i_1} & a_{i_s i_1+1} & \cdots & a_{i_s i_s-1} & a_{i_s i_s} & a_{i_s i_s+1} & \cdots & a_{i_s m+s} \\
k_{i_s+11} & \cdots & k_{i_s+1i_1-1} & a_{i_s+1i_1} & k_{i_s+1i_1+1} & \cdots & k_{i_s+1i_s-1} & a_{i_s+1i_s} & k_{i_s+1i_s+1} & \cdots & k_{i_s+1m+s} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
k_{m+s1} & \cdots & k_{m+s\,i_1-1} & a_{m+s\,i_1} & k_{m+s\,i_1+1} & \cdots & k_{m+s\,i_s-1} & a_{m+s\,i_s} & k_{m+s\,i_s+1} & \cdots & k_{m+s\,m+s}
\end{bmatrix}_{(m+s)\times(m+s)} \tag{7}
$$

The locations of the inserted rows and columns are only associated with the numberings of nodes whose supports are deleted. The objective of static reanalysis is to calculate the displacement vector **x** by making full use of the initial information, especially the Cholesky factorization of the initial stiffness matrix, so that the computational cost can be remarkably reduced. Once the displacements are achieved, the stresses can be readily determined by utilizing stress-displacement relations.

4. The static reanalysis method for modification of deleting some supports

In this section, our proposed method is first derived, then the details of computation is given, finally the efficiency of the method is studied.

4.1 The derivation of the method for the problem

Suppose **x** and **r** in Eq. (5) are the vectors with components x_i and r_i, respectively. Then Eq. (5) can be written in the following form

$$\left\{\begin{array}{l}
k_{11}x_1+\cdots+k_{1i_1-1}x_{i_1-1}+a_{1i_1}x_{i_1}+k_{1i_1+1}x_{i_1+1}+\cdots+k_{1i_s-1}x_{i_s-1}+a_{1i_s}x_{i_s}+k_{1i_s+1}x_{i_s+1}+\cdots+k_{1m+s}x_{m+s}=r_1\\
\vdots\\
k_{i_1-11}x_1+\cdots+k_{i_1-1i_1-1}x_{i_1-1}+a_{i_1-1i_1}x_{i_1}+k_{i_1-1i_1+1}x_{i_1+1}+\cdots+k_{i_1-1i_s-1}x_{i_s-1}+a_{i_1-1i_s}x_{i_s}+\\
\qquad k_{i_1-1i_s+1}x_{i_s+1}+\cdots+k_{i_1-1m+s}x_{m+s}=r_{i_1-1}\\
a_{i_11}x_1+\cdots+a_{i_1i_1-1}x_{i_1-1}+a_{i_1i_1}x_{i_1}+a_{i_1i_1+1}x_{i_1+1}+\cdots+a_{i_1i_s-1}x_{i_s-1}+a_{i_1i_s}x_{i_s}+a_{i_1i_s+1}x_{i_s+1}+\cdots+a_{i_1m+s}x_{m+s}=r_{i_1}\\
k_{i_1+11}x_1+\cdots+k_{i_1+1i_1-1}x_{i_1-1}+a_{i_1+1i_1}x_{i_1}+k_{i_1+1i_1+1}x_{i_1+1}+\cdots+k_{i_1+1i_s-1}x_{i_s-1}+a_{i_1+1i_s}x_{i_s}+\\
\qquad k_{i_1+1i_s+1}x_{i_s+1}+\cdots+k_{i_1+1m+s}x_{m+s}=r_{i_1+1}\\
\vdots\\
k_{i_s-11}x_1+\cdots+k_{i_s-1i_1-1}x_{i_1-1}+a_{i_s-1i_1}x_{i_1}+k_{i_s-1i_1+1}x_{i_1+1}+\cdots+k_{i_s-1i_s-1}x_{i_s-1}+a_{i_s-1i_s}x_{i_s}+\\
\qquad k_{i_s-1i_s+1}x_{i_s+1}+\cdots+k_{i_s-1m+s}x_{m+s}=r_{i_s-1}\\
a_{i_s1}x_1+\cdots+a_{i_si_1-1}x_{i_1-1}+a_{i_si_1}x_{i_1}+a_{i_si_1+1}x_{i_1+1}+\cdots+a_{i_si_s-1}x_{i_s-1}+a_{i_si_s}x_{i_s}+a_{i_si_s+1}x_{i_s+1}+\cdots+a_{i_sm+s}x_{m+s}=r_{i_s}\\
k_{i_s+11}x_1+\cdots+k_{i_s+1i_1-1}x_{i_1-1}+a_{i_s+1i_1}x_{i_1}+k_{i_s+1i_1+1}x_{i_1+1}+\cdots+k_{i_s+1i_s-1}x_{i_s-1}+a_{i_s+1i_s}x_{i_s}+\\
\qquad k_{i_s+1i_s+1}x_{i_s+1}+\cdots+k_{i_s+1m+s}x_{m+s}=r_{i_s+1}\\
\vdots\\
k_{m+s1}x_1+\cdots+k_{m+si_1-1}x_{i_1-1}+a_{m+si_1}x_{i_1}+k_{m+si_1+1}x_{i_1+1}+\cdots+k_{m+si_s-1}x_{i_s-1}+a_{m+si_s}x_{i_s}+\\
\qquad k_{m+si_s+1}x_{i_s+1}+\cdots+k_{m+sm+s}x_{m+s}=r_{m+s}
\end{array}\right. \qquad (8)$$

Put the variables $x_{i_1},\cdots,x_{i_s}$ at the ends of all the equations, i.e.

$$\left\{\begin{array}{l}
k_{11}x_1+\cdots+k_{1i_1-1}x_{i_1-1}+k_{1i_1+1}x_{i_1+1}+\cdots+k_{1i_s-1}x_{i_s-1}+k_{1i_s+1}x_{i_s+1}+\cdots+k_{1m+s}x_{m+s}+a_{1i_1}x_{i_1}+\cdots+a_{1i_s}x_{i_s}=r_1\\
\vdots\\
k_{i_1-11}x_1+\cdots+k_{i_1-1i_1-1}x_{i_1-1}+k_{i_1-1i_1+1}x_{i_1+1}+\cdots+k_{i_1-1i_s-1}x_{i_s-1}+k_{i_1-1i_s+1}x_{i_s+1}+\cdots+k_{i_1-1m+s}x_{m+s}+\\
\qquad a_{i_1-1i_1}x_{i_1}+\cdots+a_{i_1-1i_s}x_{i_s}=r_{i_1-1}\\
a_{i_11}x_1+\cdots+a_{i_1i_1-1}x_{i_1-1}+a_{i_1i_1+1}x_{i_1+1}+\cdots+a_{i_1i_s-1}x_{i_s-1}+a_{i_1i_s+1}x_{i_s+1}+\cdots+a_{i_1m+s}x_{m+s}+a_{i_1i_1}x_{i_1}+\cdots+a_{i_1i_s}x_{i_s}=r_{i_1}\\
k_{i_1+11}x_1+\cdots+k_{i_1+1i_1-1}x_{i_1-1}+k_{i_1+1i_1+1}x_{i_1+1}+\cdots+k_{i_1+1i_s-1}x_{i_s-1}+k_{i_1+1i_s+1}x_{i_s+1}+\cdots+k_{i_1+1m+s}x_{m+s}+\\
\qquad a_{i_1+1i_1}x_{i_1}+\cdots+a_{i_1+1i_s}x_{i_s}=r_{i_1+1}\\
\vdots
\end{array}\right.$$

$$\left\{\begin{array}{l}
\quad\vdots \\
k_{i_s-11}x_1+\cdots+k_{i_s-1i_1-1}x_{i_1-1}+k_{i_s-1i_1+1}x_{i_1+1}+\cdots+k_{i_s-1i_s-1}x_{i_s-1}+k_{i_s-1i_s+1}x_{i_s+1}+\cdots+k_{i_s-1m+s}x_{m+s}+ \\
\qquad a_{i_s-1i_1}x_{i_1}+\cdots+a_{i_s-1i_s}x_{i_s}=r_{i_s-1} \\
a_{i_s1}x_1+\cdots+a_{i_si_1-1}x_{i_1-1}+a_{i_si_1+1}x_{i_1+1}+\cdots+a_{i_si_s-1}x_{i_s-1}+a_{i_si_s+1}x_{i_s+1}+\cdots+a_{i_sm+s}x_{m+s}+a_{i_si_1}x_{i_1}+\cdots+a_{i_si_s}x_{i_s}=r_{i_s} \\
k_{i_s+11}x_1+\cdots+k_{i_s+1i_1-1}x_{i_1-1}+k_{i_s+1i_1+1}x_{i_1+1}+\cdots+k_{i_s+1i_s-1}x_{i_s-1}+k_{i_s+1i_s+1}x_{i_s+1}+\cdots+k_{i_s+1m+s}x_{m+s}+ \\
\qquad a_{i_s+1i_1}x_{i_1}+\cdots+a_{i_s+1i_s}x_{i_s}=r_{i_s+1} \\
\quad\vdots \\
k_{m+s1}x_1+\cdots+k_{m+si_1-1}x_{i_1-1}+k_{m+si_1+1}x_{i_1+1}+\cdots+k_{m+si_s-1}x_{i_s-1}+k_{m+si_s+1}x_{i_s+1}+\cdots+k_{m+sm+s}x_{m+s}+ \\
\qquad a_{m+si_1}x_{i_1}+\cdots+a_{m+si_s}x_{i_s}=r_{m+s}
\end{array}\right. \tag{9}$$

Rearrange the sequence of the above equations as follows

$$\left\{\begin{array}{l}
k_{11}x_1+\cdots+k_{1i_1-1}x_{i_1-1}+k_{1i_1+1}x_{i_1+1}+\cdots+k_{1i_s-1}x_{i_s-1}+k_{1i_s+1}x_{i_s+1}+\cdots+k_{1m+s}x_{m+s}+a_{1i_1}x_{i_1}+\cdots+a_{1i_s}x_{i_s}=r_1 \\
\quad\vdots \\
k_{i_1-11}x_1+\cdots+k_{i_1-1i_1-1}x_{i_1-1}+k_{i_1-1i_1+1}x_{i_1+1}+\cdots+k_{i_1-1i_s-1}x_{i_s-1}+k_{i_1-1i_s+1}x_{i_s+1}+\cdots+k_{i_1-1m+s}x_{m+s}+ \\
\qquad a_{i_1-1i_1}x_{i_1}+\cdots+a_{i_1-1i_s}x_{i_s}=r_{i_1-1} \\
k_{i_1+11}x_1+\cdots+k_{i_1+1i_1-1}x_{i_1-1}+k_{i_1+1i_1+1}x_{i_1+1}+\cdots+k_{i_1+1i_s-1}x_{i_s-1}+k_{i_1+1i_s+1}x_{i_s+1}+\cdots+k_{i_1+1m+s}x_{m+s}+ \\
\qquad a_{i_1+1i_1}x_{i_1}+\cdots+a_{i_1+1i_s}x_{i_s}=r_{i_1+1} \\
\quad\vdots \\
k_{i_s-11}x_1+\cdots+k_{i_s-1i_1-1}x_{i_1-1}+k_{i_s-1i_1+1}x_{i_1+1}+\cdots+k_{i_s-1i_s-1}x_{i_s-1}+k_{i_s-1i_s+1}x_{i_s+1}+\cdots+k_{i_s-1m+s}x_{m+s}+ \\
\qquad a_{i_s-1i_1}x_{i_1}+\cdots+a_{i_s-1i_s}x_{i_s}=r_{i_s-1} \\
k_{i_s+11}x_1+\cdots+k_{i_s+1i_1-1}x_{i_1-1}+k_{i_s+1i_1+1}x_{i_1+1}+\cdots+k_{i_s+1i_s-1}x_{i_s-1}+k_{i_s+1i_s+1}x_{i_s+1}+\cdots+k_{i_s+1m+s}x_{m+s}+ \\
\qquad a_{i_s+1i_1}x_{i_1}+\cdots+a_{i_s+1i_s}x_{i_s}=r_{i_s+1} \\
\quad\vdots \\
k_{m+s1}x_1+\cdots+k_{m+si_1-1}x_{i_1-1}+k_{m+si_1+1}x_{i_1+1}+\cdots+k_{m+si_s-1}x_{i_s-1}+k_{m+si_s+1}x_{i_s+1}+\cdots+k_{m+sm+s}x_{m+s}+ \\
\qquad a_{m+si_1}x_{i_1}+\cdots+a_{m+si_s}x_{i_s}=r_{m+s} \\
a_{i_11}x_1+\cdots+a_{i_1i_1-1}x_{i_1-1}+a_{i_1i_1+1}x_{i_1+1}+\cdots+a_{i_1i_s-1}x_{i_s-1}+a_{i_1i_s+1}x_{i_s+1}+\cdots+a_{i_1m+s}x_{m+s}+a_{i_1i_1}x_{i_1}+\cdots+a_{i_1i_s}x_{i_s}=r_{i_1} \\
\quad\vdots \\
a_{i_s1}x_1+\cdots+a_{i_si_1-1}x_{i_1-1}+a_{i_si_1+1}x_{i_1+1}+\cdots+a_{i_si_s-1}x_{i_s-1}+a_{i_si_s+1}x_{i_s+1}+\cdots+a_{i_sm+s}x_{m+s}+a_{i_si_1}x_{i_1}+\cdots+a_{i_si_s}x_{i_s}=r_{i_s}
\end{array}\right. \tag{10}$$

Eq. (10) is equivalent to Eq. (8), note that the coefficient matrix in Eq. (10) is $\begin{bmatrix}\mathbf{K}_0 & \mathbf{K}_{12}\\ \mathbf{K}_{12}^{\mathrm{T}} & \mathbf{K}_{22}\end{bmatrix}$, where $\mathbf{K}_{12}\in R^{m\times s}$, $\mathbf{K}_{22}\in R^{s\times s}$. The above process can be summarized at matrix-vector level as follows. There exists a permutation matrix **P** such that

$$\mathbf{PKP}^{\mathrm{T}}\mathbf{Px}=\mathbf{Pr} \tag{11}$$

where $\mathbf{PKP}^{\mathrm{T}}=\begin{bmatrix}\mathbf{K}_0 & \mathbf{K}_{12}\\ \mathbf{K}_{12}^{\mathrm{T}} & \mathbf{K}_{22}\end{bmatrix}$. Let $\mathbf{C}=\begin{bmatrix}\mathbf{K}_0 & \mathbf{K}_{12}\\ \mathbf{K}_{12}^{\mathrm{T}} & \mathbf{K}_{22}\end{bmatrix}\in R^{(m+s)\times(m+s)}$, $\bar{\mathbf{x}}=\mathbf{Px}$ and $\bar{\mathbf{r}}=\mathbf{Pr}$, then we have

$$\mathbf{C}\bar{\mathbf{x}}=\bar{\mathbf{r}} \tag{12}$$

It is important to emphasize that **C** is SPD since **K** is SPD and $\mathbf{C}=\mathbf{PKP}^{\mathrm{T}}$. Assume the Cholesky factorization of **C** is

$$\mathbf{C} = \mathbf{L}\mathbf{L}^{\mathrm{T}} \tag{13}$$

where **L** is the Cholesky triangle of matrix **C**. Let **L** be partitioned in the following form

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \tag{14}$$

where $\mathbf{L}_{11} \in R^{m\times m}$, $\mathbf{L}_{21} \in R^{s\times m}$, $\mathbf{L}_{22} \in R^{s\times s}$, and **0** denotes the corresponding zero matrix.

Recall from property 1 in Section 2, we have $\mathbf{L}_{11}=\mathbf{L}_0$. Thus, only $\mathbf{L}_{21}$ and $\mathbf{L}_{22}$ require to be calculated. The Cholesky factorization of **C** can be computed from the m+1th row, as given in the following algorithm.

Algorithm 2

For $i = m+1, \cdots, m+s$
  for $j = 1, \cdots, i-1$
    for $k = 1, \cdots, j-1$
      $c_{ij} = c_{ij} - l_{ik} l_{jk}$
    end
    $l_{ij} = c_{ij} / l_{jj}$
  end
  for $k = 1, \cdots, i-1$
    $c_{ii} = c_{ii} - l_{ik}^2$
  end
  $l_{ii} = \sqrt{c_{ii}}$
End

We name Algorithm 2 the continued Cholesky factorization algorithm. The algorithm involves $3m^2s+3ms^2+s^3$ flops and is numerical stable since it can be viewed as a part of Cholesky factorization algorithm of matrix **C**.

4.2 The details of computation

During the process of computing, we do not need to explicitly form the matrices **P** and **C**. Instead we work directly with the last sth rows of **C**. Once the Cholesky factorization of **C** is achieved, $\bar{\mathbf{x}}$ can be calculated via the forward and backward substitutions with little cost. $\mathbf{x} = \mathbf{P}^{\mathrm{T}}\bar{\mathbf{x}}$ can be obtained by the following way instead of using matrix-vector multiplication. Put the last sth entries of $\bar{\mathbf{x}}$ into the i_1th, i_2th, …, i_sth component of **x**, and the remainder of **x** is padded with the first mth entries of $\bar{\mathbf{x}}$ in order. In addition, most part of **L**, i.e., the upper-left corner is band matrix $\mathbf{L}_0$, this property can be utilized by the forward and backward substitutions for reducing the computational cost.

4.3 The efficiency of the method

The computational cost of our proposed algorithm can be quantified by the number of flops.

Assume the half-band widths of the initial stiffness matrix and the modified stiffness matrix are the same, and let b denote the half-band width. The case of $b<<m$ is studied. Recall from the last part of Section 4.1, the Cholesky factorization of **C** involves $3m^2s+3ms^2+s^3$ flops by utilizing the proposed algorithm. The forward and backward substitutions require $2mb+4ms+2s^2$ since the upper-left corner of **L** is a band matrix $\mathbf{L}_0$. Thus, the total computational cost of the proposed algorithm is $3m^2s+3ms^2+s^3+2mb+4ms+2s^2$ flops. Direct analysis method costs (m+s)(b^2+8b+1) flops (Golub and Van Loan 1996) since the Cholesky factorization of the modified stiffness matrix is required. The theoretical speed up S_t is defined as the ratio of the flops using the direct analysis method to that using the proposed method (Leu and Tsou 2000), that is

$$S_t = \frac{(m+s)(b^2+8b+1)}{3m^2s+3ms^2+s^3+2mb+4ms+2s^2} \tag{15}$$

Eq. (15) can be approximately by

$$S_t \approx \frac{m(b^2+8b)}{3m^2s+3ms^2+2mb+4ms} = \frac{b^2+8b}{3ms+3s^2+2b+4s} \tag{16}$$

since $s<<m$ and $b<<m$. From Eq. (16), it can be observed that the smaller s is, the larger S_t is. Using Eq. (16) yields $S_t \geq 1$, if

$$s \leq \frac{\sqrt{9m^2+12b^2+24m+72b+16}-3m-4}{6} \tag{17}$$

i.e., when the number of the deleted support constraints s satisfies the inequality (17), the computational cost of the proposed method is equal to or less than that of the direct analysis method.

Note that the matrix **C** may not be a band matrix, the method does not keep the band wide of the modified stiffness matrix **K**, this is a drawback of our method. Fortunately, the number of the entries in **L** which need to be calculated is small since $s<<m$.

5. Numerical examples

In this section, two examples are given to demonstrate the effectiveness of the proposed method. All the computations are completed on a PC: Pentium 4, quad-core CPU with 2.66 GHz, 2 GB RAM. Compaq Visual Fortran 6.5 is used.

Example 1

An offshore oil platform is studied in this example, as shown in Fig. 2. The material modulus of elasticity is $E=2\times10^{11}$ Pa and the Poisson’s ratio is $\upsilon=0.3$. The height of the structure is 162 m in which $h_1=15$ m and $h_2=6$ m. The length, width and thickness of the two rectangular platforms are 16 m, 12 m and 3.6×10^{-2} m, respectively. The structure is discretized into a finite element model with 396 elements and 192 nodes. Every node has 6 DOFs except the 8 constrained nodes and the total number of DOFs of the structure is 1104. Fig. 3 shows the 8 constrained nodes, where ● denotes the constrained node. Three kinds of elements are used, i.e., 40 beam elements, 96 plate elements and 260 pipe elements. All the beams are under the two rectangular platforms, as shown

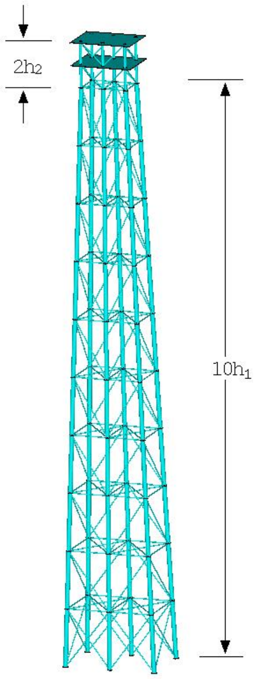


Fig. 2 An offshore oil platform

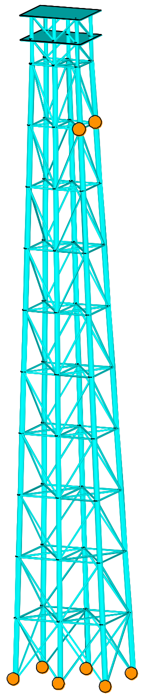


Fig. 3 The initial constrained nodes of the offshore oil platform

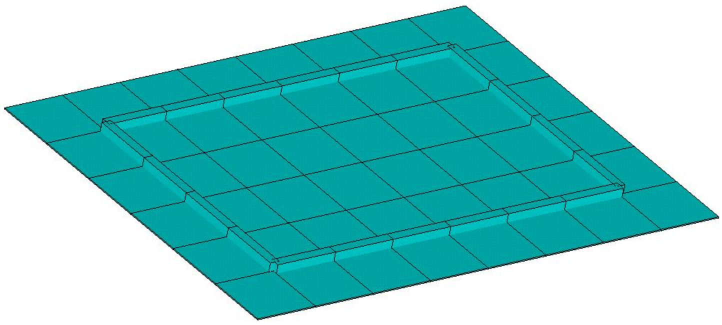


Fig. 4 The beams of the offshore oil platform

in Fig. 4. The beam cross-section is 0.3 m×0.3 m. There are three pipe cross-section sizes: outer radius 0.6 m, thickness 3×10^{-2} m; outer radius 0.4 m, thickness 2×10^{-2} m; and outer radius 0.15 m, thickness 0.1 m. The size of each plate element is 2 m×2 m, and the thickness is 3.6×10^{-2} m. Every node of the structure is subjected to a vertical load $P=-1\times10^{4}$ N. The modification is deleting the two side constrained nodes of the structure. Thus, the modified structure has 6 constrained nodes, as shown in Fig. 5. The number of DOFs of the modified structure is 1116.

Table 1 presents the 2-norm of the displacement vectors $\mathbf{x}_p$ and $\mathbf{x}_d$ of the modified structure calculated by the proposed method and direct analysis, respectively. The 2-norm of their difference is also given in this table. It can be observed that, the calculated results of the two methods are almost identical. Here, direct analysis means that the displacements vector is obtained by the

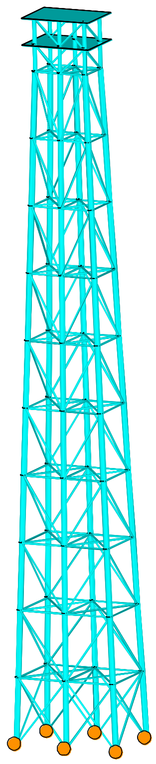


Fig. 5 The constrained nodes of the offshore oil platform after modification

Table 1 The 2-norm of the displacement vectors and their difference of the modified offshore oil platform

	$\mathbf{x}_p$	$\mathbf{x}_d$	$\mathbf{x}_p-\mathbf{x}_d$
The 2-norm	0.261101	0.261101	2.411969×10^{-13}

Table 2 The computational times for the modified offshore oil platform

	Proposed method	Direct analysis
The computational times	0.110938s	3.737500s

Cholesky factorization of the modified stiffness matrix, the forward and back substitutions. The computational times for the offshore oil platform are listed in Table 2. It can be seen that the computational time of the proposed method is much less than that of direct analysis.

Example 2

Consider the framework of a six-storey building, as shown in Fig. 6. The length, width and height of the building are 36 m, 18 m and 24 m, respectively. The height of each floor is 4 m. The material modulus of elasticity and the Poisson's ratio are E=3×10^{10} Pa and υ=0.2, respectively. A finite element model is employed to simulate the framework under a given load. The model has 918 elements and 561 nodes. Every node has 6 DOFs except the 15 constrained nodes and the total number of DOFs of the structure is 3276. All the constrained nodes are at the bottom of the

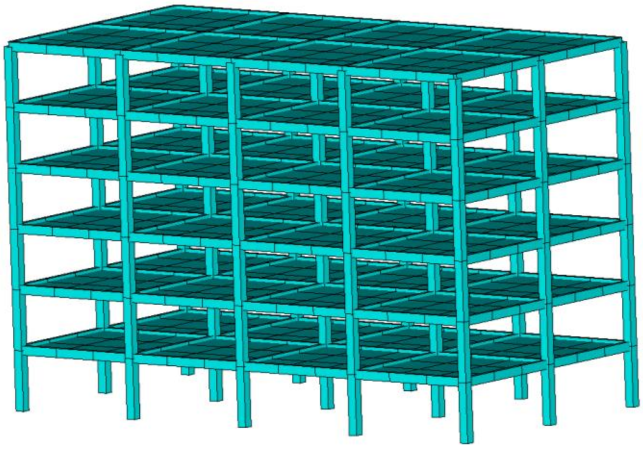


Fig. 6 The framework of a six-storey building

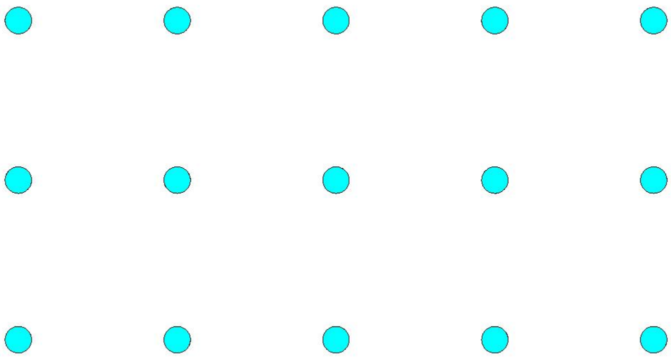


Fig. 7 The initial constrained nodes of the framework

Table 3 The 2-norm of the displacement vectors and their difference of the modified framework

	$\mathbf{x}_p$	$\mathbf{x}_d$	$\mathbf{x}_p-\mathbf{x}_d$
The 2-norm	0.030071	0.030071	4.654269×10^{-16}

structure, as showed in Fig. 7, in which '•' denotes the constrained node. The structure includes two types of elements: 486 beam elements and 432 plate elements. The thickness of each plate is 0.18 m and the size is 3 m×3 m. The beams have two cross-section sizes, i.e., 0.6 m×0.6 m (those are perpendicular to the ground) and 0.3 m×0.6 m (others). The modification is deleting the rotation constraints of three nodes denoted by 'Δ' in Fig. 8. Thus, the total number of DOFs of the modified structure is 3285. The nodes which are on the roof of the building are subjected to a vertical load $P=-2\times10^4$ N.

Table 3 gives the 2-norm of the displacement vectors $\mathbf{x}_p$, $\mathbf{x}_d$ and their difference. For the meanings of $\mathbf{x}_p$ and $\mathbf{x}_d$, see the above example. The computational times for the modified structure are given in Table 4. It is can be observed that the computational time of our proposed method is much less compared with direct analysis.

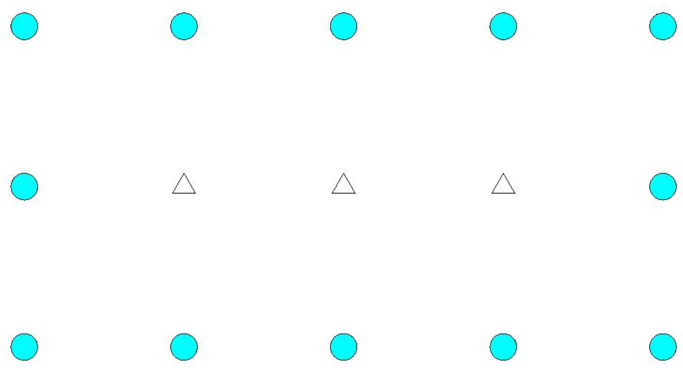


Fig. 8 The constrained nodes of the framework after modification

Table 4 The computational times for the modified framework

	Proposed method	Direct analysis
The computational times	0.982813s	15.992188s

6. Conclusions

This paper has focused on the static reanalysis problem with deleting some support constraints whose orientations are the same as the orientations of some axes of the global coordinate system. An efficient reanalysis method has been proposed. The method makes full use of the initial information. A property of Cholesky factorization is studied and utilized. An algorithm named the continued Cholesky factorization algorithm is proposed and employed for our problem. The method provides exact solutions, thus it belongs to the direct reanalysis methods. Numerical examples have shown that the calculated results of the proposed method are the same as that of direct analysis, meanwhile the computational times can be significantly reduced. However, the proposed method can only deal with a special case of support modifications. Future work is to study the static reanalysis problem for the general support modifications (adding or deleting of some support constraints, and the orientations of the added or deleted support constraints are unlimited).

Acknowledgments

The work was supported by EU FP7-IRSES projects EYE2E (269118) and the Fundamental Research Funds for the Central Universities.

References

Abu Kassim, A.M. and Topping, B.H.V. (1987), "Static reanalysis: a review", *J. Struct. Eng.*, ASCE, **113**(5), 1029-1045.

Akgün, M.A., Garcelon, J.H. and Haftka, R.T. (2001), "Fast exact linear and non-linear structural reanalysis

and the Sherman-Morrison-Woodbury formulas", *Int. J. Numer. Method. Eng.*, **50**(7), 1587-1606.
Chen, T.Y. and Huang, J.H. (2013), "An efficient and practical approach to obtain a better optimum solution for structural optimization", *Eng. Optim.*, **45**(8), 1005-1026.
Davis, T.A. (2006), *Direct Methods for Sparse Linear System*, SIAM, Philadelphia.
Golub, G.H. and Van Loan, C.F. (1996), *Matrix Computations*, 3rd Edition, The Johns Hopkins University Press, Baltimore, London.
Kirsch, U. (2008), *Reanalysis of Structures*, Springer, Dordrecht.
Leu, L.J. and Tsou, C.H. (2000), "Application of a reduction method for reanalysis to nonlinear dynamic analysis of framed structures", *Comput. Mech.*, **26**(5), 497-505.
Li, Z.G. and Wu, B.S. (2007), "A preconditioned conjugate gradient approach to structural reanalysis for general layout modifications", *Int. J. Numer. Method. Eng.*, **70**(5), 505-522.
Liu, C.H., Cheng, I., Tsai, A.C., Wang, L.J. and Hsu, J.Y. (2010), "Using multiple point constraints in finite element analysis of two dimensional contact problems", *Struct. Eng. Mech.*, **36**(1), 95-110.
Liu, H.F., Wu, B.S., Lim, C.W. and Li, Z.G. (2012a), "An approach for structural static reanalysis with unchanged number of degrees of freedom", *Struct. Multidisc. Optim.*, **45**(5), 681-692.
Liu, H.F., Wu, B.S. and Li, Z.G. (2012b), "An efficient approach to structural static reanalysis with added support constraints", *Struct. Eng. Mech.*, **43**(3), 273-285.
Olhoff, N. and Taylor, J.E. (1983), "On structural optimization", *J. Appl. Mech.*, ASME, **50**(4), 1139-1151.
Pais, M.J., Yeralan, S.N., Davis, T.A. and Kim, N.H. (2012), "An exact reanalysis algorithm using incremental Cholesky factorization and its application to crack growth modeling", *Int. J. Numer. Method. Eng.*, **91**(12), 1358-1364.
Takezawa, A., Nishiwaki, S., Izui, K. and Yoshimura, M. (2006), "Structural optimization using function-oriented elements to support conceptual designs", *J. Mech. Des*., ASME, **128**(4), 689-700.
Tanskanen, P. (2006), "A multiobjective and fixed elements based modification of the evolutionary structural optimization method", *Comput. Method. Appl. M.*, **196**(1-3), 76-90.
Terdalkar, S.S. and Rencis, J.J. (2006), "Graphically driven interactive finite element stress reanalysis for machine elements in the early design stage", *Finit. Elem. Anal. Des.*, **42**(10), 884-899.
Wang, D., Jiang, J.S. and Zhang, W.H. (2004), "Optimization of support positions to maximize the fundamental frequency of structures", *Int. J. Numer. Method. Eng.*, **61**(10), 1584-1602.
Zhu, J.H. and Zhang, W.H. (2010), "Integrated layout design of supports and structures", *Comput. Method. Appl. M.*, **199**(9-12), 557-569.

CC