Struct Multidisc Optim (2012) 45:681–692
DOI 10.1007/s00158-011-0723-y

RESEARCH PAPER

An approach for structural static reanalysis with unchanged number of degrees of freedom

H. F. Liu · B. S. Wu · C. W. Lim · Z. G. Li

Received: 27 October 2010 / Revised: 2 July 2011 / Accepted: 30 September 2011 / Published online: 9 November 2011

Abstract This paper presents an approach for structural static reanalysis with unchanged number of degrees of freedom. Preconditioned conjugate gradient method is employed, and a new preconditioner is constructed by updating the Cholesky factorization of the initial stiffness matrix with little cost. The proposed method preserves the ease of implementation and significantly improves the quality of the results. In particular, the accuracy of the approximate solutions can adaptively be monitored. Numerical examples show that the condition number of preconditioned system using the new preconditioner is much smaller than that using the initial stiffness matrix as the preconditioner. Therefore, the fast convergence and accurate results can be obtained by the proposed approach.

Keywords Structural static reanalysis · Preconditioned conjugate gradient (PCG) method · Cholesky factorization · Rank-one modification

H. F. Liu
Department of Mathematics, School of Science,
Xi'an Jiaotong University, Xi'an 710049,
People's Republic of China

B. S. Wu (✉) · Z. G. Li
Department of Mechanics and Engineering Science,
School of Mathematics, Jilin University, Changchun
130012, People's Republic of China
e-mail: wubaisheng@yahoo.com

C. W. Lim
Department of Building and Construction,
City University of Hong Kong,
Tat Chee Avenue, Kowloon, Hong Kong,
People's Republic of China

1 Introduction

It is well known that structures are progressively modified during the process of design or optimization. Each modification requires a fresh analysis for displacements and stresses, and the process involves repeated and tremendous calculations. This results in extensive studies on structural static reanalysis (Abu Kasim and Topping 1987; Barthelemy and Haftka 1993). The purpose of static reanalysis is to utilize the information of the original structure to accurately evaluate structural responses under a given static load of the structure after modifications without repeatedly solving the complete system of the modified equilibrium equations so that the computational cost can be significantly reduced (Abu Kasim and Topping 1987).

To date, many static reanalysis methods have been developed, especially for the case of fixed layout structural modifications where the number of degrees of freedom (DOFs) keeps unchanged. Generally speaking, these reanalysis methods can be divided into the following two categories: direct methods and approximate methods. Direct methods give exact closed-form solutions and are suitable for cases where the changes in design variables are large in magnitude, yet only affect a relative small number of elements. Most of these methods update the inverse of the modified stiffness matrix using Sherman-Morrison-Woodbury formulae (Sherman and Morrison 1949; Woodbury 1950). Various improvements were proposed and the readers are referred to Akgün et al. (2001). Direct methods are inefficient when there are changes in many elements of the stiffness matrix.

Approximate reanalysis methods (Kirsch 2002) provide approximate solutions of the response of the modified structure using the information obtained during the full analysis of the original structure. These methods are applicable to modifications where the changes in design variables are small in magnitude, yet may significantly influence a large portion or the entire stiffness matrix. Generally, the approximate methods can be divided into the following four classes: local approximations, global approximations, combined approximations (CA), and preconditioned conjugate gradient (PCG) approximations. Local approximations (Kirsch 2002) (also called single-point approximations) use the information obtained at a single design point. Methods such as the Taylor series expansion or the binomial series expansion about a given design point belong to local approximations. These methods are effective only in cases which involve small changes in the design variables. For large changes in the design variables, accuracy of the approximate results often deteriorates and may become meaningless. Global approximations (Kirsch 2002) (also called multipoint approximations) include polynomial fitting, response surface and reduced basis methods. These methods are obtained by analyzing the structure at a number of design points, and they are valid for the whole space (or, at least, large regions of it). However, global approximations may require much computational cost in problems with a large number of design variables. The CA method attempts to give global qualities to local approximations (Kirsch 2002). The idea of the CA method is combining the reduced basis method with the first terms of a series expansion. Similar to local approximations, the calculations are based on results of a single exact analysis. Each subsequent reanalysis requires the solution of a reduced system. The reduced system is further uncoupled by using the Gram-Schmidt orthonormalization procedure. However, the CA method requires calculating and storing all basis vectors, forming the reduced stiffness matrix or completing the Gram-Schmidt orthonormalization for the basis vectors, etc. Certainly, these procedures will increase the computational cost and may result in numerical instability. The PCG method (Golub and Van Loan 1996; Saad 1996) is the most critical ingredient in the development of efficient solutions for challenging problems in engineering and science computation (Benzi 2002). It is the most efficient method for structural static reanalysis among the approximate methods mentioned above (Kirsch et al. 2002; Wu and Li 2006; Li and Wu 2007; Wu et al. 2004). This method converts the modified system of equilibrium equations into a related system with a small condition number so that the total number of iterations required for solving the system to within some specified tolerances is significantly reduced. The procedure is easy to be implemented and only a small number of vectors need to be preserved in the memory. The accuracy of approximate solutions can adaptively be monitored. For fixed layout modifications of structural static reanalysis, the equivalence between the PCG methods of using the initial stiffness matrix as the preconditioner and the CA method has been proved (Kirsch et al. 2002; Nair 2002). For large modifications, however, the PCG method which utilizes the initial stiffness matrix as the preconditioner may converge slowly. In order to accelerate the solution convergence, the successive matrix inversion method (Bae and Grandhi 2004; Bae et al. 2006a) is exploited to construct the preconditioner (Bae et al. 2006b). Since the incremental stiffness matrix is divided by columns of interest, the preconditioner may be non-symmetric and indefinite. For this reason, GMRES method (Saad and Schultz 1986) or Bi-CGSTAB method (Van der Vorst 1992) has to be employed for structural static reanalysis problems, but breakdown or stagnation may take place.

In this paper, a new preconditioner is constructed for fixed layout modifications with unchanged number of DOFs by updating the Cholesky factorization of the original stiffness matrix. Compared with the PCG method using the initial stiffness matrix as the preconditioner, the condition number for the proposed PCG method using the new preconditioner is considerably reduced and fast convergence can therefore be achieved.

2 Reanalysis problem formulation

Structural static reanalysis for the fixed layout modifications with unchanged number of DOFs can be stated as follows Kirsch (1991). Given an initial design, the corresponding stiffness matrix is $\mathbf{K}_0 \in R^{n\times n}$. The displacement vector $\mathbf{y}_0$ can be calculated by the equilibrium equations

$$\mathbf{K}_0\mathbf{y}_0 = \mathbf{R}_0 \tag{1}$$

where $\mathbf{R}_0$ is the load vector. The stiffness matrix $\mathbf{K}_0$ is symmetric and positive definite (SPD). From the initial analysis, the Cholesky factorization of $\mathbf{K}_0$ has already been known

$$\mathbf{K}_0 = \mathbf{L}_0\mathbf{D}_0\mathbf{L}_0^{\mathrm{T}} \tag{2}$$

where $\mathbf{L}_0$ is a lower triangular matrix with unit elements on the diagonal, $\mathbf{D}_0$ is a diagonal matrix, $\mathbf{L}_0^{\mathrm{T}}$ denotes the transpose of $\mathbf{L}_0$. Assume a change in the design so that the modified stiffness matrix is

$$\mathbf{K} = \mathbf{K}_0 + \mathbf{\Delta}\mathrm{K} \tag{3}$$

where $\mathbf{\Delta}\mathrm{K}$ is the change of the stiffness matrix due to a change in the design and is called the incremental stiffness matrix, and $\mathbf{K} \in R^{n\times n}$ is a SPD matrix. The purpose of structural static reanalysis is to obtain efficient and accurate approximate solutions of the modified displacement vector $\mathbf{y}$ without directly solving the modified equilibrium equations

$$\mathbf{K}\mathbf{y} = \mathbf{R} \tag{4}$$

where $\mathbf{R}$ is the modified load vector which may be different from the load vector of the initial design $\mathbf{R}_0$.

Once the displacement vector $\mathbf{y}$ is obtained, the stresses can be readily determined by utilizing explicit stress-displacement relations.

3 The construction of preconditioner for PCG methods

The PCG method (Golub and Van Loan 1996; Saad 1996) will be applied to solve the present structural static reanalysis problem. The PCG method for fixed layout modifications (Kirsch 2002) is firstly reviewed, the rank-one decompositions of the stiffness matrix and the incremental stiffness matrix are then given, and a new preconditioner is finally constructed in this section.

For the case of fixed layout modifications where the number of DOFs remains unchanged, the preconditioner may be chosen as the initial stiffness matrix $\mathbf{K}_0$, and the reanalysis procedure can be stated as follows Kirsch (2002), Golub and Van Loan (1996).

Algorithm 1

```
Assume initial guess $\mathbf{x} = \mathbf{0}$
$k = 0$
$\mathbf{r} = \mathbf{R}$
$\delta_0 = \mathbf{r}^{\mathrm{T}}\mathbf{r}$
while ($\sqrt{\delta_k} > \varepsilon$)
    Solve $\mathbf{K}_0\mathbf{z} = \mathbf{r}$ for $\mathbf{z}$
    $\rho_{k+1} = \mathbf{r}^{\mathrm{T}}\mathbf{z}$
    $k = k + 1$
    if $k = 1$
      $\mathbf{p} = \mathbf{z}$
    else
      $\beta = \frac{\rho_k}{\rho_{k-1}}$
      $\mathbf{p} = \mathbf{z} + \beta\mathbf{p}$
    end
    $\mathbf{w} = \mathbf{K}\mathbf{p}$
    $\alpha = \frac{\rho_k}{\mathbf{p}^{\mathrm{T}}\mathbf{w}}$
    $\mathbf{x} = \mathbf{x} + \alpha\mathbf{p}$
    $\mathbf{r} = \mathbf{r} - \alpha\mathbf{w}$
    $\delta_k = \mathbf{r}^{\mathrm{T}}\mathbf{r}$
end
$\mathbf{y} = \mathbf{x}$
```

where ε is a specified error tolerance. It should be noted that the solution of $\mathbf{K}_0\mathbf{z} = \mathbf{r}$ is inexpensive by utilizing the Cholesky factorization in (2).

The convergent rate of PCG method can be estimated by the following inequality (Golub and Van Loan 1996):

$$\left\|\mathbf{x}_k - \mathbf{x}^*\right\|_2 \le 2\sqrt{\kappa}\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \left\|\mathbf{x}_0 - \mathbf{x}^*\right\|_2 \tag{5}$$

where $\mathbf{x}_k$, $\mathbf{x}_0$ and $\mathbf{x}^*$ represent the kth approximate solution, the initial guess and the exact solution, respectively, κ is the spectral condition number of the coefficient matrix of the preconditioned system, which is the ratio of maximum and minimum eigenvalues of its coefficient matrix. From the estimation (5), it can be seen that the smaller κ is, the faster the speed of the convergence of PCG method is. Thus, the purpose of PCG method is to reduce the spectral condition number κ so that the computational cost and the computational time can be significantly reduced.

Note that each of the stiffness matrices $\mathbf{K}_0$ and $\mathbf{K}$ can be decomposed into a series of rank-one matrices, the incremental stiffness matrix $\boldsymbol{\Delta}\mathbf{K} = \mathbf{K} - \mathbf{K}_0$ can therefore be written into sum of a series of rank-one matrices. The linear isotropic three-dimensional beam elements will be used to validate this decomposition.

For beam element mentioned above, the element stiffness matrix $[\mathbf{K}^{(\mathrm{e})}]$ is a sum of four matrices (Rao 2004): the axial stiffness matrix $[\mathbf{K}_{axial}]$, the bending stiffness matrix $[\mathbf{K}_{bending}]_{xy}$ in xy plane, the bending stiffness matrix $[\mathbf{K}_{bending}]_{xz}$ in xz plane, and the torsional stiffness matrix $[\mathbf{K}_{torsion}]$:

$$\left[\mathbf{K}^{(\mathrm{e})}\right] = [\mathbf{K}_{axial}] + \left[\mathbf{K}_{bending}\right]_{xy} + \left[\mathbf{K}_{bending}\right]_{xz} + [\mathbf{K}_{torsion}] \tag{6}$$

where

$$[\mathbf{K}_{axial}] = \mathbf{u}_1\mathbf{u}_1^{\mathrm{T}}, \tag{7}$$

$$[\mathbf{K}_{torsion}] = \mathbf{u}_2\mathbf{u}_2^{\mathrm{T}}, \tag{8}$$

$$\left[\mathbf{K}_{bending}\right]_{xy} = \mathbf{u}_3\mathbf{u}_3^{\mathrm{T}} + \mathbf{u}_4\mathbf{u}_4^{\mathrm{T}}, \tag{9}$$

$$\left[\mathbf{K}_{bending}\right]_{xz} = \mathbf{u}_5\mathbf{u}_5^{\mathrm{T}} + \mathbf{u}_6\mathbf{u}_6^{\mathrm{T}}, \tag{10}$$

$$\mathbf{u}_1 = \sqrt{\frac{EA}{L}}\left[1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ -1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\right]^{\mathrm{T}}, \tag{11}$$

$$\mathbf{u}_2 = \sqrt{\frac{GJ}{L}}\left[0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ -1\ \ 0\ \ 0\right]^{\mathrm{T}}, \tag{12}$$

$$\mathbf{u}_3 = \frac{1}{L}\sqrt{\frac{EI_z}{L}} \times\left[0\ 2\sqrt{3}\ 0\ 0\ 0\ \sqrt{3}L\ 0\ -2\sqrt{3}\ 0\ 0\ 0\ \sqrt{3}L\right]^{\mathrm{T}}, \tag{13}$$

$$\mathbf{u}_4 = \sqrt{\frac{EI_z}{L}}\left[0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ -1\right]^{\mathrm{T}}, \tag{14}$$

$$\mathbf{u}_5 = \frac{1}{L}\sqrt{\frac{EI_y}{L}} \times\left[0\ 0\ 2\sqrt{3}\ 0\ -\sqrt{3}L\ 0\ 0\ 0\ -2\sqrt{3}\ 0\ -\sqrt{3}L\ 0\right]^{\mathrm{T}}, \tag{15}$$

$$\mathbf{u}_6 = \sqrt{\frac{EI_y}{L}}\left[0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ -1\ \ 0\right]^{\mathrm{T}}, \tag{16}$$

and L denotes the length of beam element, E represents the material elastic modulus, A is the element cross-sectional area, I_z and I_y denote the moments of inertia about z axis and y axis, respectively, G represents the material shear modulus, J denotes the torsional moment of inertia of the cross-sectional area.

Substituting (7)–(10) into (6), the decomposition of the element stiffness matrix can be obtained:

$$\left[\mathbf{K}^{(\mathrm{e})}\right] = \sum_{i=1}^{6} \mathbf{u}_i \mathbf{u}_i{}^{\mathrm{T}} \tag{17}$$

For other elements such as truss, plate and shell etc., similar decomposition of the element stiffness matrix can also be established.

It is noted that the decomposition of the element stiffness matrix of the linear isotropic three-dimensional beam given in (17) is carried out with respect to the local xyz coordinate system. After the transformation of the coordinates and the extension of the size, all the element stiffness matrices are employed to form the global stiffness matrix. Therefore, the global stiffness matrix can be decomposed into a series of rank-one matrices:

$$\overline{\mathbf{K}} = \sum_{s=1}^{p}\left[\overline{\overline{\mathbf{K}}}{}_s^{(\mathrm{e})}\right] = \sum_{s=1}^{p}\sum_{i=1}^{6} \overline{\overline{\mathbf{u}}}_{is}\overline{\overline{\mathbf{u}}}{}_{is}^{\mathrm{T}},\ \ \overline{\overline{\mathbf{u}}}_{is} = \mathbf{G}_s^{\mathrm{T}}\mathbf{R}_s^{\mathrm{T}}\mathbf{u}_{is} \tag{18}$$

where p is the total number of elements, $\mathbf{G}_s$ represents the transforming matrix of the sth element which expands the element stiffness matrix, by means of the connectivity relations, into part of the global stiffness matrix, $\mathbf{R}_s$ denotes the matrix of the coordinates transformation of the sth element, and $\mathbf{u}_{is}$ $(i = 1, \cdots\cdots, 6)$ is given in (11)–(16). For the details of $\mathbf{G}_s$ and $\mathbf{R}_s$, we refer readers to Rao (2004). Application of displacement boundary conditions yields the modified global stiffness matrix, which can also be decomposed into a series of rank-one matrices.

In Gill et al. (1974), the algorithm of updating the Cholesky factorization for a matrix following a rank-one modification was presented. Assuming $\mathbf{A} \in R^{n\times n}$ is a SPD matrix, the Cholesky factorization of $\mathbf{A}$ is given by

$$\mathbf{A} = \mathbf{LDL}^{\mathrm{T}} \tag{19}$$

where $\mathbf{L}$ is a unit lower triangular matrix and $\mathbf{D}$ is a diagonal matrix. Further assume that $\overline{\mathbf{A}}$ is formed by a symmetric rank-one modification of matrix $\mathbf{A}$, i.e.,

$$\overline{\mathbf{A}} = \mathbf{A} + \eta\mathbf{ww}^{\mathrm{T}}\ (\eta \in R,\ \eta \neq 0) \tag{20}$$

where $\mathbf{w} \in R^n$ is a column vector. The Cholesky factorization of $\overline{\mathbf{A}}$ can be obtained by the following way instead of directly factoring $\overline{\mathbf{A}}$

$$\begin{aligned}\overline{\mathbf{A}} &= \mathbf{A} + \eta\mathbf{ww}^{\mathrm{T}}\\ &= \mathbf{L}\left(\mathbf{D} + \eta\mathbf{pp}^{\mathrm{T}}\right)\mathbf{L}^{\mathrm{T}}\end{aligned} \tag{21}$$

where $\mathbf{Lp} = \mathbf{w}$, and $\mathbf{p}$ is calculated by a forward substitution. It is necessary to assume that $\overline{\mathbf{A}}$ is SPD due to the existence of Cholesky factorization. Note that $\mathbf{D} + \eta\mathbf{pp}^{\mathrm{T}}$ is SPD if and only if $\overline{\mathbf{A}}$ is SPD. The Cholesky factorization $\mathbf{D} + \eta\mathbf{pp}^{\mathrm{T}} = \hat{\mathbf{L}}\hat{\mathbf{D}}\hat{\mathbf{L}}^{\mathrm{T}}$ can be obtained with little cost by utilizing the following property: the subdiagonal elements of every column of $\hat{\mathbf{L}}$ are multiples of the corresponding elements of the vector $\mathbf{p}$. The property can be proved by induction. Explicitly forming the first column of $\hat{\mathbf{L}}\hat{\mathbf{D}}\hat{\mathbf{L}}^{\mathrm{T}}$, we obtain the equations

$$\hat{d}_1 = d_1 + \eta p_1^2 \tag{22}$$

$$\hat{\mathbf{L}}_{i1}\hat{d}_1 = \eta p_1 p_i, \quad i = 2, \ \cdots, n \tag{23}$$

From (23), we have

$$\hat{\mathbf{L}}_{i1} = p_i \frac{\eta p_1}{\hat{d}_1}, \quad i = 2, \ \cdots, n \tag{24}$$

which shows that the subdiagonal elements of the first column of $\hat{\mathbf{L}}$ are same multiple of the corresponding elements of $\mathbf{p}$. Assume the first $(j\text{–}1)$th columns are of the form

$$\hat{\mathbf{L}}_{ik} = p_i\beta_k, \quad k = 1, \ \cdots, j-1; \quad i = k+1, \ \cdots, n \tag{25}$$

The jth column of $\hat{\mathbf{L}}$ can be determined by the following equations

$$\sum_{i=1}^{j-1} \hat{\mathbf{L}}_{ji}^2 \hat{d}_i + \hat{d}_j = d_j + \eta p_j^2 \tag{26}$$

$$\sum_{i=1}^{j-1} \hat{\mathbf{L}}_{ji}\hat{\mathbf{L}}_{si}\hat{d}_i + \hat{\mathbf{L}}_{sj}\hat{d}_j = \eta p_j p_s, \quad s = j+1, \ \cdots, n \tag{27}$$

Substituting (25) into (26) and (27) gives

$$p_j^2 \sum_{i=1}^{j-1} \beta_i^2 \hat{d}_i + \hat{d}_j = d_j + \eta p_j^2 \tag{28}$$

$$p_j p_s \sum_{i=1}^{j-1} \beta_i^2 \hat{d}_i + \hat{\mathbf{L}}_{sj}\hat{d}_j = \eta p_j p_s, \quad s = j+1, \ \cdots, n \tag{29}$$

From the last equation, we have

$$\hat{\mathbf{L}}_{sj} = p_s \frac{p_j}{\hat{d}_j}\left(\eta - \sum_{i=1}^{j-1}\beta_i^2\hat{d}_i\right) = p_s\beta_j, \quad s = j+1, \cdots, n \tag{30}$$

where $\beta_j = \frac{p_j}{\hat{d}_j}\left(\eta - \sum_{i=1}^{j-1}\beta_i^2\hat{d}_i\right)$. Hence, the subdiagonal elements of the jth column of $\hat{\mathbf{L}}$ are same multiple of the corresponding elements of the vector $\mathbf{p}$. Consequently, the above property has been proved by induction. The property can be also employed to calculate the product of $\mathbf{L}$ and $\hat{\mathbf{L}}$ inexpensively which is subsequently required:

$$\bar{\mathbf{A}} = \mathbf{L}\hat{\mathbf{L}}\hat{\mathbf{D}}\hat{\mathbf{L}}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}} = \overline{\mathbf{L}}\,\overline{\mathbf{D}}\,\overline{\mathbf{L}}^{\mathrm{T}} \tag{31}$$

where $\overline{\mathbf{L}} = \mathbf{L}\hat{\mathbf{L}}$, $\overline{\mathbf{D}} = \hat{\mathbf{D}}$. The product of $\mathbf{L}$ and $\hat{\mathbf{L}}$ can be obtained by the following Algorithm.

```
1. Define $\mathbf{w}^{(1)} = \mathbf{w}$
2. For $j = 1, \ldots, n$
     for $i = j+1, \ldots, n$
       $\mathbf{w}_i^{(j+1)} = \mathbf{w}_i^{(j)} - p_j\mathbf{L}_{ij}$
       $\overline{\mathbf{L}}_{ij} = \mathbf{L}_{ij} + \beta_j\mathbf{w}_i^{(j+1)}$
     end
   End
```

Hence, the algorithm of updating the Cholesky factorization for a matrix following a rank-one modification can be given as follows, see Gill et al. (1974) for details.

Algorithm 2

```
1. Define $\eta_1 = \eta$, $\mathbf{w}^{(1)} = \mathbf{w}$.
2. For $j = 1, 2, \cdots, n$, compute
     $p_j = w_j^{(j)}$
     $\overline{d_j} = d_j + \eta_j p_j^2$
     $\beta_j = \frac{p_j\eta_j}{\overline{d_j}}$
     $\eta_{j+1} = \frac{d_j\eta_j}{\overline{d_j}}$
     for $r = j+1, \cdots, n$
       $\mathbf{w}_r^{(j+1)} = \mathbf{w}_r^{(j)} - p_j\mathbf{L}_{rj}$
       $\overline{\mathbf{L}}_{rj} = \mathbf{L}_{rj} + \beta_j\mathbf{w}_r^{(j+1)}$
     end
   End
```

The number of operations necessary to compute the modified factorization using Algorithm 2 is $2n^2 + O(n)$ flops, and is roughly equal to the cost of a matrix-vector product when n is large enough. For static reanalysis problem, components of the correction vector $\mathbf{w}$ usually contain many "0"s and the stiffness matrix has a banded form. The two properties can be applied to the algorithm of updating the Cholesky factorization for the purpose of reducing the computational cost. Let $2b+1$ be the bandwidth of $\overline{\mathbf{A}}$ and t satisfy the following relationship:

$$\mathbf{w} = (0, \cdots, 0, w_t, \cdots, w_n)^{\mathrm{T}}, \quad \mathrm{w}_t \neq 0 \tag{32}$$

Then, the changes of elements of the matrix $\mathbf{L}$ only occur in the lower right corner, which is a $(n - t + 1) \times (n - t + 1)$ submatrix. Other elements remain unchanged and are unnecessary to be re-calculated. The property of band matrix can be easily utilized to reduce the computational cost of the algorithm since some zero elements of the matrix $\overline{\mathbf{L}}$ can be determined before implementing the algorithm. Owing to the above two properties, the number of operations of the algorithm can be reduced. Let $f(n,t,b)$ be the number of the flops of the algorithm, then

$$f(n,t,b) = \begin{cases} 2b(2n-2t+2-b) + O(n-t+1), & n-t+1 > b \\ 2(n-t+1)^2 + O(n-t+1), & n-t+1 \le b \end{cases} \tag{33}$$

Generally speaking, t and b usually satisfy $t > 1$ and $b < n$ and the computational cost for the algorithm of updating the Cholesky factorization is less than the cost of a matrix-vector product. The algorithm will be utilized to construct the preconditioner for structural static reanalysis problems mentioned above.

The construction of the preconditioner should be based on the following principles (Van der Vorst 2003):

1. The preconditioner is a good approximation to the coefficient matrix of the linear system in some sense.
2. The cost of the construction of the preconditioner is not prohibitive.
3. The preconditioned system is much easier to solve than the original system.

Based on the rules above, the choice of preconditioners for fixed layout modifications of structures in this paper is described as follows. The modification in a design optimization is generally determined by sensitivity information. For each iteration, the information for objective and constraint functions with respect to design variables is used to improve the current design. Based on the sensitivity information, major and minor contributing variables which impose large and small modifications on the current structural design, respectively, can be determined. It is obvious that the major contributing design variables have a large influence on the structural behavior. The proposed method will separate the effect of the major contributing design variables on the structural behavior at the coefficient matrix level.

Based on the discussion above, the incremental stiffness matrix can be written in the following form

$$\boldsymbol{\Delta}\mathbf{K} = \boldsymbol{\Delta}\mathbf{K}_1 + \boldsymbol{\Delta}\mathbf{K}_2 \tag{34}$$

where $\boldsymbol{\Delta}\mathbf{K}_1$ represents the contributions of elements where changes in design variables are significant in magnitude, yet only a relative small number of elements are involved, $\boldsymbol{\Delta}\mathbf{K}_2$ denotes the contribution of elements where changes in design variables are small in magnitude, but a relative large number of elements are affected. The optimal division of $\boldsymbol{\Delta}\mathbf{K}$ into $\boldsymbol{\Delta}\mathbf{K}_1$ and $\boldsymbol{\Delta}\mathbf{K}_2$ should further be studied. Note that $\boldsymbol{\Delta}\mathbf{K}_1$ is only a part of the incremental stiffness matrix and it can be decomposed into a series of rank-one matrices, while $\boldsymbol{\Delta}\mathbf{K}_2$ does not need such a decomposition. In this paper, $\mathbf{K}_0 + \boldsymbol{\Delta}\mathbf{K}_1$ is selected as the preconditioner of the PCG method instead of $\mathbf{K}_0$. It is necessary to require that $\mathbf{K}_0 + \boldsymbol{\Delta}\mathbf{K}_1$ is a SPD matrix. The proposed preconditioned matrix $\mathbf{K}_0 + \boldsymbol{\Delta}\mathbf{K}_1$ approximates the modified stiffness matrix $\mathbf{K}_0 + \boldsymbol{\Delta}\mathbf{K}$ than $\mathbf{K}_0$ better. Algorithm 2 is applied to obtain the Cholesky factorization of $\mathbf{K}_0 + \boldsymbol{\Delta}\mathbf{K}_1$ by considering a rank-one modification each time without directly factoring $\mathbf{K}_0 + \boldsymbol{\Delta}\mathbf{K}_1$. Therefore, the cost of constructing the preconditioner is inexpensive. Once the Cholesky factorization of $\mathbf{K}_0 + \boldsymbol{\Delta}\mathbf{K}_1$ is obtained, the preconditioner system with coefficient matrix $\mathbf{K}_0 + \boldsymbol{\Delta}\mathbf{K}_1$ can be solved with little cost. Finally, replacing the original preconditioner $\mathbf{K}_0$ by the present $\mathbf{K}_0 + \boldsymbol{\Delta}\mathbf{K}_1$ in Algorithm 1, we can calculate the displacements of the modified structure.

4 Numerical examples

In this section, three examples are presented to illustrate the efficiency of the proposed method. An error tolerance ε of 10^{-6} is fixed in the three examples. The proposed method is compared with the PCG method of using the initial stiffness matrix $\mathbf{K}_0$ as the preconditioner, since the latter is the most efficient method for fixed layout modifications of structural static reanalysis among the approximate methods. The computations of Example 1 are completed on a PC: Pentium 4, quad-core CPU with 2.66 GHz, 2 GB RAM. MATLAB 7.2.0.232(R2006a) is used. The computations of Examples 2 and 3 are completed on a PC: Pentium 4, 3.2 GHz CPU, 2 GB RAM. The compiler is Compaq Visual Fortran 6.5.

Example 1 Consider a truss structure as shown in Fig. 1. The elasticity modulus of material is $E = 2 \times 10^{11}$ Pa, the cross-sectional area of all the elements is 1×10^{-3} m^2. The length and width of the structure are 2 m and 1 m, respectively. The structure has 6 nodes, where 4 nodes are located at the vertices of a 2 m × 1 m rectangle, the remaining 2 nodes are located at the midpoints of its long edges. The eight unknowns are the horizontal and vertical displacements at nodes 3, 4, 5 and 6, since nodes 1 and 2 are constrained. Nodes 3 and 5 are subjected to the vertical loads, $P_1 = P_2 = 2 \times 10^3$ N. The stiffness matrix $\mathbf{K}_0$ of the structure is given as follows:

$$\mathbf{K}_0 = 10^8 \times \begin{bmatrix} 5.41421356 & 0 & 0 & 0 & -2 & 0 & -0.70710678 & -0.70710678 \\ 0 & 3.41421356 & 0 & -2 & 0 & 0 & -0.70710678 & -0.70710678 \\ 0 & 0 & 4 & 0 & 0 & 0 & -2 & 0 \\ 0 & -2 & 0 & 2 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & -2 \\ -0.70710678 & -0.70710678 & -2 & 0 & 0 & 0 & 2.70710678 & 0.70710678 \\ -0.70710678 & -0.70710678 & 0 & 0 & 0 & -2 & 0.70710678 & 2.70710678 \end{bmatrix}$$

The load vector of the structure is $\mathbf{P} = [0\ -2000\ 0\ 0\ 0\ -2000\ 0\ 0]^{\mathrm{T}}$. For a modification, the cross-sectional areas of four elements are increased from originally 1×10^{-3} m^2 to 1.1×10^{-3} m^2, and one truss element with cross-sectional area 1.1×10^{-3} m^2 is added, as shown in Fig. 2. The stiffness matrix $\mathbf{K}$ of the modified structure can be given as follows:

$$\mathbf{K} = 10^8 \times \begin{bmatrix} 5.68492424 & 0.07071068 & 0 & 0 & -2.2 & 0 & -0.77781746 & -0.77781746 \\ 0.07071068 & 3.48492424 & 0 & -2 & 0 & 0 & -0.77781746 & -0.77781746 \\ 0 & 0 & 4.97781745 & -0.77781746 & -0.77781746 & 0.77781746 & -2.2 & 0 \\ 0 & -2 & -0.77781746 & 2.77781745 & 0.77781746 & -0.77781746 & 0 & 0 \\ -2.2 & 0 & -0.77781746 & 0.77781746 & 2.97781745 & -0.77781746 & 0 & 0 \\ 0 & 0 & 0.77781746 & -0.77781746 & -0.77781746 & 2.97781745 & 0 & -2.2 \\ -0.77781746 & -0.77781746 & -2.2 & 0 & 0 & 0 & 2.97781745 & 0.77781746 \\ -0.77781746 & -0.77781746 & 0 & 0 & 0 & -2.2 & 0.77781746 & 2.97781745 \end{bmatrix}$$

The incremental stiffness matrix $\mathbf{\Delta K}$ is

$$\Delta\mathbf{K} = 10^8 \times \begin{bmatrix} 0.27071068 & 0.07071068 & 0 & 0 & -0.2 & 0 & -0.07071068 & -0.07071068 \\ 0.07071068 & 0.07071068 & 0 & 0 & 0 & 0 & -0.07071068 & -0.07071068 \\ 0 & 0 & 0.97781745 & -0.77781746 & -0.77781746 & 0.77781746 & -0.2 & 0 \\ 0 & 0 & -0.77781746 & 0.77781745 & 0.77781746 & -0.77781746 & 0 & 0 \\ -0.2 & 0 & -0.77781746 & 0.77781746 & 0.97781745 & -0.77781746 & 0 & 0 \\ 0 & 0 & 0.77781746 & -0.77781746 & -0.77781746 & 0.97781745 & 0 & -0.2 \\ -0.07071068 & -0.07071068 & -0.2 & 0 & 0 & 0 & 0.27071067 & 0.07071068 \\ -0.07071068 & -0.07071068 & 0 & 0 & 0 & -0.2 & 0.07071068 & 0.27071067 \end{bmatrix}$$

The matrix can be divided into two parts. One is the stiffness matrix of the newly added truss element and the other is the change in the stiffness matrix that results from an increase of truss cross-sectional areas. It is obvious that the number of newly added element is one and the new element contributes significantly to the incremental stiffness matrix. Thus, the incremental stiffness matrix resulting from the newly added element is chosen as $\Delta\mathbf{K}_1$:

$$\Delta\mathbf{K}_1 = 10^8 \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.77781745 & -0.77781746 & -0.77781746 & 0.77781746 & 0 & 0 \\ 0 & 0 & -0.77781746 & 0.77781745 & 0.77781746 & -0.77781746 & 0 & 0 \\ 0 & 0 & -0.77781746 & 0.77781746 & 0.77781745 & -0.77781746 & 0 & 0 \\ 0 & 0 & 0.77781746 & -0.77781746 & -0.77781746 & 0.77781745 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

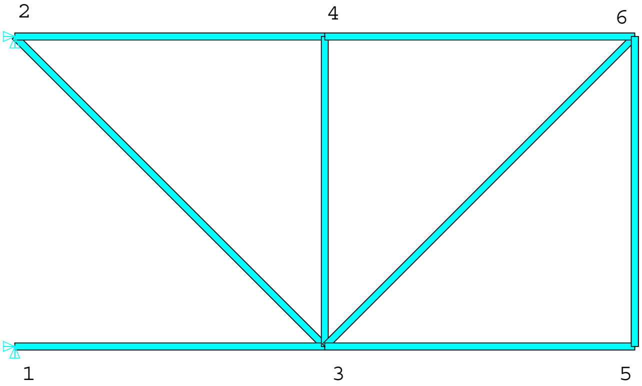

Fig. 1 A truss structure

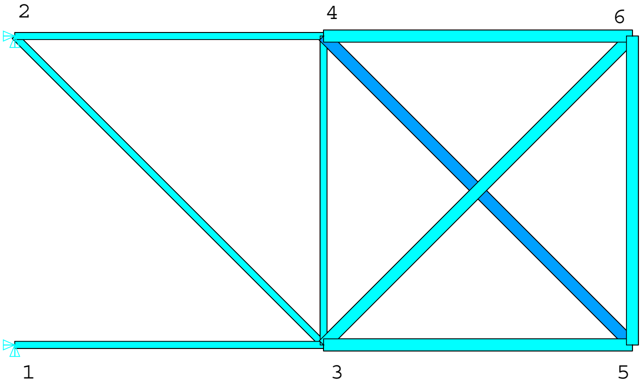

Fig. 2 A truss structure after modification

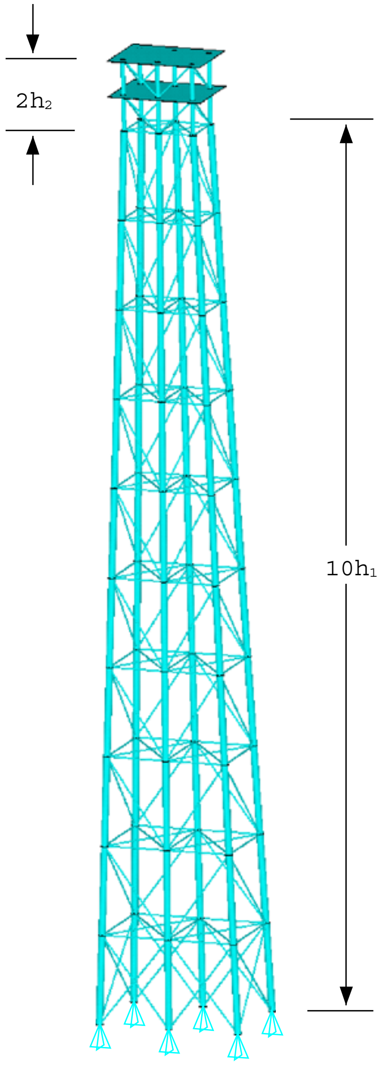

Fig. 3 An offshore oil platform

Table 1 Computational costs of the modified truss structure

Preconditioner	Iters	CCCPs	Matvecs	CN
$\mathbf{K}_0$	4	0	8	3.741449
$\mathbf{K}_0 + \Delta\mathbf{K}_1$	3	1	7	1.100000

Table 2 Computational times of the modified truss structure

Preconditioner	Time1	Time2	The total time
$\mathbf{K}_0$	0 s	0.00147 s	0.00147 s
$\mathbf{K}_0 + \Delta\mathbf{K}_1$	0.00012 s	0.00110 s	0.00122 s

It can be easily verified that $\Delta\mathbf{K}_1 = \mathbf{u}\mathbf{u}^{\mathrm{T}}$, where $\mathbf{u} = 8.81939601 \times 10^3 \times \begin{bmatrix} 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$.

Thus, the Cholesky factorization of $\mathbf{K}_0 + \Delta\mathbf{K}_1$ can be obtained by executing Algorithm 2 one time. For the problem, the computational cost of Algorithm 2 is roughly equal

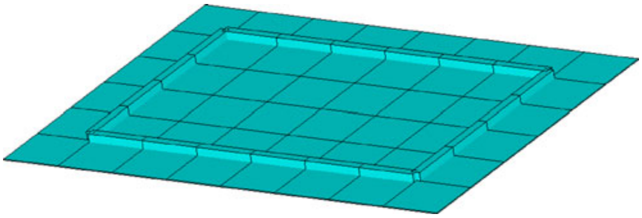

Fig. 4 The original ribs at the bottom of the platform

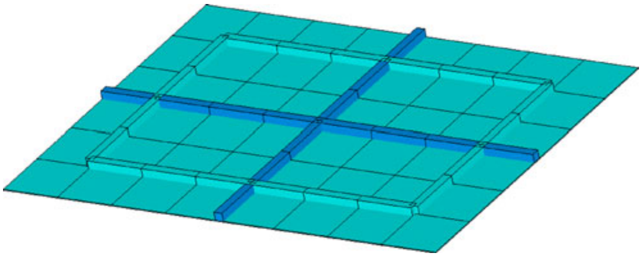

Fig. 5 The modified ribs at the bottom of the platform

Table 3 Computational costs of the modified offshore oil platform

Preconditioner	Iters	CCCPs	Matvecs	CN
$\mathbf{K}_0$	210	0	420	2.618476×10^5
$\mathbf{K}_0 + \Delta\mathbf{K}_1$	9	168	186	1.587963

Table 4 Computational times of the modified offshore oil platform

Preconditioner	Time1	Time2	The total time
$\mathbf{K}_0$	0 s	11.36094 s	11.36094 s
$\mathbf{K}_0 + \Delta\mathbf{K}_1$	0.58125 s	0.48906 s	1.07031 s

to that of a matrix-vector product. For simplicity, the computational cost of implementing Algorithm 2 one time is regarded as the same as that of a matrix-vector product. Once the Cholesky factorization of $\mathbf{K}_0+\Delta\mathbf{K}_1$ is obtained, it is selected as the preconditioner of the PCG methods instead of $\mathbf{K}_0$. The computational cost and the corresponding condition numbers are presented in Table 1, where Iters indicate the number of iterations of the PCG methods for an error tolerance $\varepsilon = 10^{-6}$, CCCPs represent the computational cost of constructing the preconditioner which is counted in terms of the number of matrix-vector multiplications, Matvecs denote the number of matrix-vector multiplications and CN is the abbreviation of the condition number.

Note that implementing each iteration of the PCG methods requires solving the preconditioner system once and one matrix-vector multiplication. The computational cost of solving the preconditioned system is equal to that of one matrix-vector multiplication when the Cholesky factorization of the preconditioner has already been known. Thus, the cost of an iteration of PCG methods is roughly equal to that of two matrix-vector multiplications.

It can be observed from Table 1 that the condition number of using $\mathbf{K}_0 + \Delta\mathbf{K}_1$ as preconditioner is smaller than that of using $\mathbf{K}_0$ as preconditioner. Therefore, the number of iterations of using $\mathbf{K}_0 + \Delta\mathbf{K}_1$ as preconditioner is fewer than that of using $\mathbf{K}_0$ as preconditioner. However, some extra computational cost is required to obtain the Cholesky factorization of the new preconditioner. Integrating these calculations altogether, one may find that the total computational cost of using $\mathbf{K}_0 + \Delta\mathbf{K}_1$ as the preconditioner is less than that of using $\mathbf{K}_0$ when the same error tolerance $\varepsilon = 10^{-6}$ is specified.

The computational times for the modified truss structure are given in Table 2. Here, Time1 denotes the computational time of constructing the preconditioner, Time2 represents the computational time for executing the PCG iterations. Note that the Cholesky factorization of $\mathbf{K}_0$ has already been known from the initial analysis, thus the computational time of constructing the preconditioner is 0 s when $\mathbf{K}_0$ is used as the preconditioner.

It is observed from Table 2 that the total computational time of using $\mathbf{K}_0 + \Delta\mathbf{K}_1$ as the preconditioner is less than that of using $\mathbf{K}_0$ when the same error tolerance $\varepsilon = 10^{-6}$ is specified.

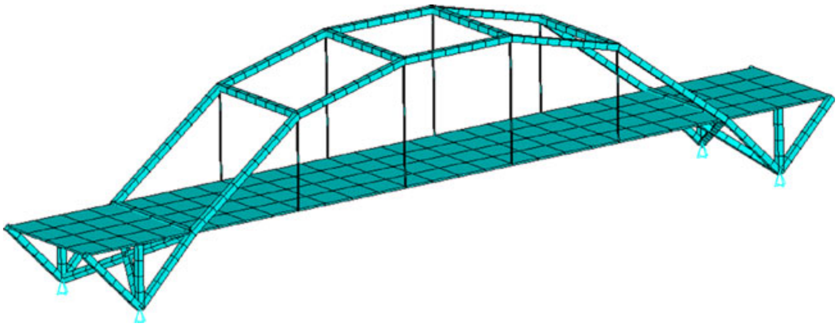

Fig. 6 The original suspension-arch bridge

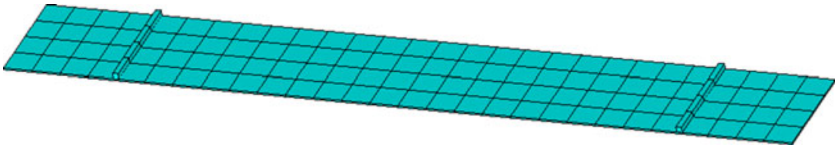

Fig. 7 The original ribs under the bridge

Example 2 Consider an offshore oil platform as shown in Fig. 3. The material modulus of elasticity is $E = 2 \times 10^{11}$ Pa and the Poisson's ratio is $\upsilon = 0.3$. The height of the structure is 162 m where $h_1 = 15$ m and $h_2 = 6$ m. The bottom of the structure is constrained by 6 nodes at boundary of a 24 m × 16 m rectangle where 4 nodes are located at the vertices of the rectangle, the remaining 2 nodes are located at the midpoints of its long edges. Twelve pipes are connected to the boundaries of two 12 m × 8 m rectangles where 8 nodes are located at the vertices of these rectangles, the other 4 nodes are located at the midpoints of their long sides. The length, width and thickness of the two rectangular platforms are 16 m, 12 m and 3×10^{-2} m, respectively. The offshore oil platform is discretized into a finite element model with 396 elements and 192 nodes. Every node has 6 DOFs except the 6 constrained nodes and the total number of DOFs of the structure is 1116. Three kinds of elements have been employed, they include: 40 beam elements, 96 plate elements and 260 pipe elements. 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. Every node of the structure is subjected to a vertical load: for nodes of the two platforms $P_1 = 5 \times 10^4$ N, for the remaining nodes $P_2 = 3 \times 10^4$ N. In order to reinforce the two platforms (modification), one rib with 0.3 m × 0.3 m cross-section is added at the bottom of every platform, respectively, and the thickness of the two platforms is increased from 3×10^{-2} m to 3.5×10^{-2} m. The newly added ribs have been discretized into beam elements. Thus, 28 beam elements are added in the structure and the thickness of all plate elements is changed. Figures 4 and 5 show the initial ribs and the ribs after modifications at the bottom of the platform, respectively.

The incremental stiffness matrix is composed of stiffness matrices of newly added beam elements and those resulting from the increase of platform thickness. It is obvious that the number of newly added beam elements is small and its contribution to incremental stiffness matrix is great. Thus, the part of incremental stiffness matrix resulting from the newly added beam elements is chosen as $\Delta\mathbf{K}_1$, while that from a change of platform thickness is viewed as $\Delta\mathbf{K}_2$. Since $\Delta\mathbf{K}_1$ can easily be decomposed into a series of rank-one matrices, the Cholesky factorization of $\mathbf{K}_0 + \Delta\mathbf{K}_1$ can be obtained by executing 168 times Algorithm 2. For the problem, the computational cost of Algorithm 2 is usually less than the cost of a matrix-vector product. For simplicity, the computational cost of implementing Algorithm 2 one time is regarded as the same as that of a matrix-vector product. Once the Cholesky factorization of $\mathbf{K}_0 + \Delta\mathbf{K}_1$ is obtained, the linear system with coefficient matrix $\mathbf{K}_0 + \Delta\mathbf{K}_1$ is easy to be solved and it can be employed as a preconditioner instead of $\mathbf{K}_0$. The computational costs and the corresponding condition numbers are listed in Table 3, see Example 1 for the meanings of the column headers in the table.

For the detailed process of calculating the computational cost, see Example 1. It can be observed from Table 3 that the condition number of using $\mathbf{K}_0 + \Delta\mathbf{K}_1$ as preconditioner is much smaller than that of using $\mathbf{K}_0$ as preconditioner. Therefore, the number of iterations of using $\mathbf{K}_0 + \Delta\mathbf{K}_1$ as preconditioner is fewer than that of using $\mathbf{K}_0$ as preconditioner. However, some extra computational cost is required to obtain the Cholesky factorization of the new preconditioner.

The computational time for the modified offshore oil platform is given in Table 4. For the meanings of Time1 and Time2 in Table 4, see Example 1.

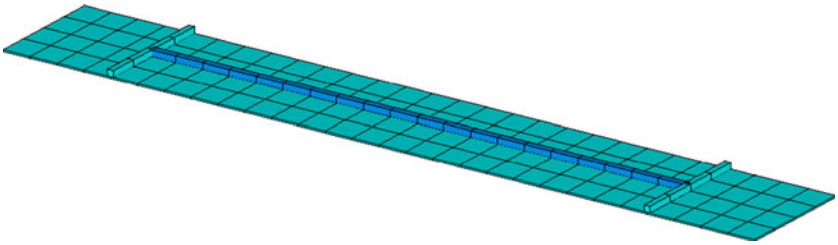

Fig. 8 The modified ribs under the bridge

It is obvious from Table 4 that the total computational time of using $\mathbf{K}_0 + \Delta\mathbf{K}_1$ as the preconditioner is less than that of using $\mathbf{K}_0$ when the same error tolerance $\varepsilon = 10^{-6}$ is specified.

Example 3 The suspension-arch bridge shown in Fig. 6 is studied in this example. The length and width of the bridge are 140 m and 20 m, respectively. The distance from the support points to the bridge surface is 10 m and the height of the arch is 20 m. A finite element model is established to simulate the bridge under a given load. The model has 356 elements and 355 nodes. Every node has 6 DOFs except the 4 constrained nodes and the total number of DOFs of the structure is 2106. It includes three kinds of elements: 8 truss elements, 236 beam elements and 112 plate elements. The modulus of elasticity for the beam and truss is $E_1 = 2 \times 10^{11}$ Pa and the modulus of elasticity for the plate is $E_2 = 3 \times 10^{10}$ Pa. The Poisson's ratios for the beam and plate are $\upsilon_1 = 0.3$ and $\upsilon_2 = 0.2$, respectively. The cross-section of the beam is 1 m × 1 m. The cross-sectional area of the truss members is 0.0314 m^2 and the thickness of the plate is 0.3 m. Every node of the bridge surface is subjected to a vertical load $\text{P} = 1 \times 10^4$ N. To reinforce the bridge (modification), the cross-section of all beams has been increased from originally 1 m × 1 m to 1.1 m × 1.1 m, and one new 1.1 m × 1.1 m beam has been added at the bottom of the bridge surface. The new added beam has been discretized into 20 elements and the total number of the elements for the modified structure is 376. Figures 7 and 8 show the initial ribs and the ribs after modifications at the bottom of the suspension-arch bridge, respectively. It is obvious that the number of newly added beam elements is small and they contribute significantly to the incremental stiffness matrix. Thus, the incremental stiffness matrix resulting from newly added beam elements is chosen as $\Delta\mathbf{K}_1$, while that from a change of cross-sections of all original beams is viewed as $\Delta\mathbf{K}_2$. Since $\Delta\mathbf{K}_1$ can easily be decomposed into a series of rank-one matrices, Algorithm 2 is implemented 120 times to obtain the Cholesky factorization of $\mathbf{K}_0 + \Delta\mathbf{K}_1$. Then, $\mathbf{K}_0 + \Delta\mathbf{K}_1$ is selected as the preconditioner of the PCG methods instead of $\mathbf{K}_0$. The computational cost is presented in Table 5.

Table 5 Computational costs of the modified suspension-arch bridge

Preconditioner	Iters	CCCPs	Matvecs	CN
$\mathbf{K}_0$	1133	0	2266	3.888313×10^6
$\mathbf{K}_0 + \Delta\mathbf{K}_1$	11	120	142	1.464102

Table 6 Computational times of the modified suspension-arch bridge

Preconditioner	Time 1	Time 2	The total time
$\mathbf{K}_0$	0 s	241.60000 s	241.60000 s
$\mathbf{K}_0 + \Delta\mathbf{K}_1$	0.52188 s	2.350000 s	2.87188 s

From Table 5, it can be seen that the total computational cost of using $\mathbf{K}_0 + \Delta\mathbf{K}_1$ is less than that of using the preconditioner $\mathbf{K}_0$ when an error tolerance $\varepsilon = 10^{-6}$ is given. The computational times using preconditioners $\mathbf{K}_0$ and $\mathbf{K}_0 + \Delta\mathbf{K}_1$ are shown in Table 6. It is obvious from Table 6 that the computational time of using $\mathbf{K}_0 + \Delta\mathbf{K}_1$ as the preconditioner is much less than that of using $\mathbf{K}_0$.

5 Conclusions

An efficient PCG approach for structural static reanalysis with unchanged number of degrees of freedom has been presented. The approach is based on a new preconditioner constructed by updating the Cholesky factorization of the original stiffness matrix. The proposed procedure not only preserves the ease of implementation but also requires little computational cost. Fast convergences in numerical examples have been observed. The precondition technique is specially efficient in dealing with cases where small parts of elements are significantly modified while their major parts are slightly modified. In particular, when the number of the modified elements is small, a direct method can be established by utilizing the procedure of the construction of the preconditioner. Furthermore, once the Cholesky factorization of the modified stiffness matrix involving major contributing variables only is calculated, it can be used as the initial information when the structure is further modified. The future work is to generalize the proposed approach to structural static reanalysis problems with varying number of DOFs and to combine it with structural optimization.

Acknowledgments The work was partially supported by the National Natural Science Foundation of China (Grant No. 11072085) and the Fundamental Research Funds for the Central Universities.

References

Abu Kasim AM, Topping BHV (1987) Static reanalysis: a review. ASCE J Struct Eng 113(5):1029–1045

Akgün MA, Garcelon JH, Haftka RT (2001) Fast exact linear and nonlinear structural reanalysis and the Sherman-Morrison-Woodbury formulas. Int J Numer Methods Eng 50(7):1587–1606

Bae HR, Grandhi RV (2004) Successive matrix inversion method for reanalysis of engineering structural systems. AIAA J 42(8):1529–1535

Bae HR, Grandhi RV, Canfield RA (2006a) Efficient successive reanalysis technique for engineering structures. AIAA J 44(8):1883–1889

Bae HR, Grandhi RV, Canfield RA (2006b) Accelerated engineering design optimization using successive matrix inversion method. Int J Numer Methods Eng 66(9):1361–1377

Barthelemy JFM, Haftka RT (1993) Recent advances in approximation concepts for optimum structural design. In: Rozvany GIN (ed) Optimization of large structural systems. Kluwer, Dordrecht, pp 235–256

Benzi M (2002) Preconditioning techniques for large linear systems: a survey. J Comput Phys 182(2):418–477

Gill PE, Golub GH, Murray W, Saunders MA (1974) Methods for modifying matrix factorizations. Math Comput 28(126):505–535

Golub GH, Van Loan CF (1996) Matrix computations, 3rd edn. The Johns Hopkins University Press, Baltimore

Kirsch U (1991) Reduced basis approximations of structural displacements for optimal design. AIAA J 29(10):1751–1758

Kirsch U (2002) Design-oriented analysis of structures. Kluwer Academic, Dordrecht

Kirsch U, Kocvara M, Zowe J (2002) Accurate reanalysis of structures by a preconditioned conjugate gradient method. Int J Numer Methods Eng 55(2):233–251

Li ZG, Wu BS (2007) A preconditioned conjugate gradient approach to structural reanalysis for general layout modifications. Int J Numer Methods Eng 70(5):505–522

Nair PB (2002) Equivalence between the combined approximations technique and Krylov subspace methods. AIAA J 40(5):1021–1023

Rao SS (2004) The finite element method in engineering, 4th edn. Butterworth-Heinemann, Boston

Saad Y (1996) Iterative methods for sparse linear system. PWS Publishing Company, Boston

Saad Y, Schultz MH (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comp 7(3):856–869

Sherman J, Morrison WJ (1949) 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 20(4):621

Van der Vorst HA (1992) Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J Sci Stat Comp 13(2):631–644

Van der Vorst HA (2003) Iterative Krylov methods for large linear systems. Cambridge University Press, Cambridge

Woodbury M (1950) Inverting modified matrices. Memorandum report 42, Statistical Research Group, Princeton University, Princeton

Wu BS, Li ZG (2006) Static reanalysis of structures with added degrees of freedom. Commun Numer Methods Eng 22(4):269–281

Wu BS, Lim CW, Li ZG (2004) A finite element algorithm for reanalysis of structures with added degrees of freedom. Finite Elem Anal Des 40(13–14):1791–1801