An aggregation multilevel iterative solver with shift acceleration for eigenvalue analysis of large-scale structures

Sergiy Yu. Fialko
Department of Structural Mechanics
S.P.Timoshenko Institute of Mechanics of NAS of Ukraine, P. Nesterov str. 3, 03057 Kiev, Ukraine
e-mail: string@inmech.kiev.ua

Abstract

The report presents a modified preconditioned conjugate gradient (MPCG) method based on an aggregated multilevel preconditioning with shifts, intended for solution of large-scale natural vibration problems resulting from the FEM application. The application of a properly chosen shift essentially improves the capability of the preconditioning to predict low vibration modes and accelerates the convergence. In many cases the use of the shift allows one to avoid the convergence lock which occurs when a conventional preconditioned conjugate gradient method (PCG) is applied. A modified iterative algorithm has been developed because each modification of the shift changes spectral properties of the preconditioning and destroys previously accumulated conjugate directions. The efficiency of the MPCG approach is illustrated by examples.

Keywords: preconditioning, aggregation approach, multilevel, shift, preconditioned conjugate gradient method

1. Introduction

The eigenvalue analysis of large-scale problems is a task of great importance. Usually block modifications of subspace iteration or Lanczos methods [11], [12] are applied in commercial software for this purpose. However, when the size of a problem is very large (200 000 - 700 000 degrees of freedom and more), these methods become too expensive. The factoring of the stiffness matrix and multiple forward-backward solutions require a lot of computation time and huge disk space storage. Multiple forward-backward resolutions during the eigenvalue extraction procedure are very computation-intensive too. In this situation the application of methods which allow one to avoid the expensive factoring procedure and forward-backward substitutions is very desirable.

Iterative methods based on a direct minimization of the Rayleigh quotient do not require the stiffness matrix to be factored. Preconditioned conjugate gradient methods for solution of natural vibration problems are considered in [11]. It is shown that the incomplete Cholesky conjugate gradient approach is a powerful tool for eigenvalue analysis.

The disadvantage of the PCG method is a lock of the convergence in some cases [13].

An original approach based on coupling of the subspace iteration idea with the preconditioned gradient method is considered in [1]. An aggregation multilevel preconditioning is applied to accelerate the convergence.

A multi-grid Davidson approach is presented in [5]. The shift technique leads to an acceleration of the convergence and allows one to avoid the lock of it.

This report presents a preconditioned conjugate gradient method with shifts in the aggregation multilevel preconditioning.

2. An aggregation multilevel preconditioning

The application of the finite element method to a natural vibration problem of structural mechanics leads to the following algebraic generalized eigenvalue problem:

$$\mathbf{K}\varphi - \lambda\mathbf{M}\varphi = 0 \tag{1}$$

It is well known from the theory of iterative methods that lower modes converge worse than higher ones. The worse the problem's conditioning, the slower the convergence. The preconditioning $\mathbf{B}$ is applied to reduce the conditioning number and accelerate the convergence [3], [10], [11]:

$$\mathbf{A}\phi - \lambda\mathbf{C}\phi = 0, \quad \mathbf{A} = \mathbf{B}^{-1}\mathbf{K}, \quad \mathbf{C} = \mathbf{B}^{-1}\mathbf{M} \tag{2}$$

It is a powerful tool to fight the ill conditioning. The main idea of the multilevel approaches is to create a coarse level model which is to predict lower modes of vibration and accelerate the convergence [3], [10].

The preconditioned conjugate gradient methods based on a multilevel preconditioning combine the advantages of PCG methods with a coarse level correction. Therefore such methods keep a robust convergence even in ill-conditioned problems.

The basic idea of the multilevel preconditioning for the PCG method used in this report is presented below.

First, a coarse level model is created. Then, the restriction-prolongation operators $\mathbf{Q}^T, \mathbf{Q}$ are formulated to establish an interface between the coarse and fine level models. The procedure presented below is applied instead of an explicit solution

$$\mathbf{B}\mathbf{z}_k = \mathbf{r}_k \tag{3}$$

where $\mathbf{r}_k$ is a residual vector, k (an iteration number) will be omitted.

- Restriction of the $\mathbf{r}$ vector to the coarse level: $\mathbf{r}_f \mapsto \mathbf{r}_c$. This procedure consists of transforming the fine level model into the coarse level: $\mathbf{r}_c = \mathbf{Q}^T\mathbf{r}_f$ and $\mathbf{Q}^T$ is the restriction operator. The upper subscript ^{T} denotes a transposition, lower subscripts f, r refer to respective fine and coarse level models.
- Resolution of $\mathbf{K}_c\mathbf{z}_c = \mathbf{r}_c$, where $\mathbf{K}_c$ is the projection of the original stiffness matrix $\mathbf{K}$ onto the coarse level ($\mathbf{K}_c$ is

already decomposed and the size of the coarse level problem allows the implementation of the direct methods).

- Prolongation $\mathbf{z}_c \mapsto \mathbf{z}_f^*$ from the coarse level to the fine level. This operation consists of a reversed transformation from the coarse level model into the fine level: $\mathbf{z}_f^* = \mathbf{Q}\mathbf{z}_c$ and $\mathbf{Q}$ is the prolongation operator
- Smoothing of the vector $\mathbf{z}_f^* \mapsto \mathbf{z}_f$ after the prolongation. Rapidly fluctuating residuals appear during the prolongation. An internal iteration procedure is applied to damp the residuals.

The aggregation approach has been proposed in [1], [2]. The implementation of the aggregation approach presented here is based on element-by-element technique used to prepare the coarse level matrix $\mathbf{K}_c$ promptly. A more efficient PCG algorithm and element-by-element aggregation scheme [6], [7], [8], [9] allows us to improve the robustness of the method and incorporate it in the Robot Millennium commercial software (www.robobat.com). Now this method is incorporated in the SCAD software (www.scadgroup.com).

The aggregation approach consists of an introduction of additional connections (rigid links) to decrease the number of degrees of freedom of a given design model. The coarse level model is derived as shown below (Fig.1). Thus, the original finite-element model (fine level) is transformed into a mechanical system (coarse level), which consists of non-overlapped local rigid aggregates coupled by elastic connections. The rigid aggregates are rigid bodies due to the imposed rigid links. All nodes of the finite-element model should be included in the rigid aggregates. It is possible to treat a single node as a limit case of a minimal rigid aggregate. It is not admissible for any node to be included into more than one aggregate.

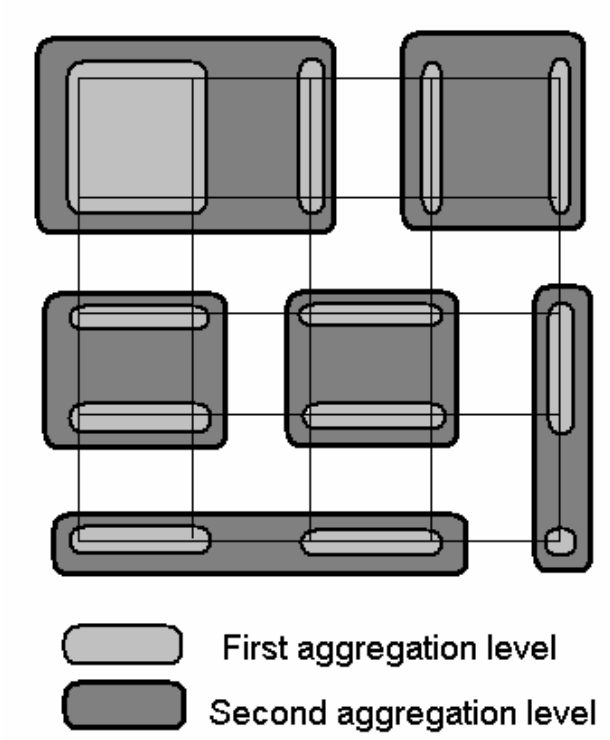


Fig.1 First and second aggregation levels for a rectangular plate with the finite element mesh 4×4

The first aggregation step is performed in an element-by-element loop. We take the first finite element and couple all nodes belonging to it. Aggregated nodes are marked. Then we take the second element and couple the remaining (unmarked) nodes. And so on.

The second aggregation level (and all the following) is performed in the same way. The aggregates from the previous level are considered to be generalized nodes. Each aggregate from the previous level which is coupled into a new aggregate of the current level, is marked to avoid a total coupling of the entire structure.

This aggregation procedure is being applied until the size of the coarsest level model becomes small enough for a direct solution. This approach keeps the topological similarity of each aggregation level to the original model (fine level).

The details are presented in [6], [7], [8], [9].

3. Introduction of a shift into the preconditioning

The application of gradient methods to the problem (1) is based on a minimization of the Rayleigh quotient

$$R(\mathbf{x}_k) = \frac{(\mathbf{K}\mathbf{x}_k, \mathbf{x}_k)}{(\mathbf{M}\mathbf{x}_k, \mathbf{x}_k)} \tag{4}$$

where $\mathbf{x}_k$ is an eigenvector approximation at k iteration step. The line search procedure reduces the minimization of (4) to the following:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k \tag{5}$$

where $\alpha_k, \mathbf{g}_k$ are an optimization parameter, obtained from minimization (4), and a gradient vector respectively.

$$\mathbf{g}_k = \frac{2}{(\mathbf{M}\mathbf{x}_k, \mathbf{x}_k)}(\mathbf{K}\mathbf{x}_k - \lambda_k \mathbf{M}\mathbf{x}_k) \tag{6}$$

The normalization $(\mathbf{M}\mathbf{x}_k, \mathbf{x}_k) = 1$ is applied, and (5), (6) become

$$\mathbf{g}_k = 2 \cdot (\mathbf{K}\mathbf{x}_k - \lambda_k \mathbf{M}\mathbf{x}_k) \tag{7}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - 2\alpha_k(\mathbf{K}\mathbf{x}_k - \lambda_k \mathbf{M}\mathbf{x}_k) = [\mathbf{I} - 2\alpha_k(\mathbf{K} - \lambda_k \mathbf{M})]\mathbf{x}_k \tag{8}$$

From (7), (8) follows:

$$\mathbf{g}_{k+1} = 2(\mathbf{K} - \lambda\mathbf{M})\mathbf{x}_{k+1} = 2(\mathbf{K} - \lambda\mathbf{M})(\mathbf{x}_k - 2\alpha_k \mathbf{g}_k) = \\ = [\mathbf{I} - 2\alpha_k(\mathbf{K} - \lambda_i \mathbf{M})]\mathbf{g}_k \tag{9}$$

and:

$$\mathbf{x}_{k+1} = [\mathbf{I} - 2\alpha_k(\mathbf{K} - \lambda_k \mathbf{M})]\mathbf{x}_k \tag{10}$$

$$\mathbf{g}_{k+1} = [\mathbf{I} - 2\alpha_k(\mathbf{K} - \lambda_k \mathbf{M})]\mathbf{g}_k \tag{11}$$

For the preconditioned problem (2) the expressions (10), (11) are:

$$\mathbf{x}_{k+1} = [\mathbf{I} - 2\alpha_k \mathbf{B}^{-1}(\mathbf{K} - \lambda_k \mathbf{M})]\mathbf{x}_k \tag{12}$$

$$\mathbf{g}_{k+1} = [\mathbf{I} - 2\alpha_k \mathbf{B}^{-1}(\mathbf{K} - \lambda_k \mathbf{M})]\mathbf{g}_k \tag{13}$$

Property 1 If $\mathbf{B} \to \mathbf{K} - \lambda_1 \mathbf{M}$, where λ_1 is a minimal eigenvalue of (1), then (12) tends to the inverse iteration procedure with the shift.

Proof. Let us suggest that $\tilde{\lambda} = \lambda_1 - \delta$, $\mathbf{B} = \mathbf{K} - \tilde{\lambda}\mathbf{M}$ and the iteration process converges to the first eigenpair. It is possible to take $\lambda_k \to \lambda_1$ for sufficiently great k.

$$\begin{aligned}
&\mathbf{x}_{k+1} = [\mathbf{I} - 2\alpha_k \mathbf{B}^{-1}(\mathbf{K} - \lambda_1 \mathbf{M})]\mathbf{x}_k = \\
&= [\mathbf{I} - 2\alpha_k \mathbf{B}^{-1}(\mathbf{K} - \lambda_1 \mathbf{M} + \delta\mathbf{M} - \delta\mathbf{M})]\mathbf{x}_k = \\
&= [\mathbf{I} - 2\alpha_k \mathbf{B}^{-1}(\mathbf{K} - \tilde{\lambda}\mathbf{M} - \delta\mathbf{M})]\mathbf{x}_k = \\
&= [\mathbf{I} - 2\alpha_k \mathbf{B}^{-1}(\mathbf{K} - \tilde{\lambda}\mathbf{M}) + 2\alpha_k \delta\mathbf{B}^{-1}\mathbf{M})]\mathbf{x}_k = \\
&= (1 - 2\alpha_k)\mathbf{I}\mathbf{x}_k + 2\alpha_k \delta(\mathbf{K} - \tilde{\lambda}\mathbf{M})^{-1}\mathbf{M}\mathbf{x}_k
\end{aligned} \tag{14}$$

Let us take $\alpha_k = 0.5$ and

$$(\mathbf{K} - \tilde{\lambda}\mathbf{M})\mathbf{x}_{k+1} = \mathbf{M}\mathbf{x}_k \tag{15}$$

This is a single step of a shifted inverse iteration method. It is well-known that the inverse iteration method with a properly selected shift has a cubic convergence.

The proof of property 1 is also presented in [9].

Property 2 Let the preconditioning be $\mathbf{B} = \mathbf{K} - \tilde{\lambda}\mathbf{M}$ where $\tilde{\lambda} = \lambda_1 - \delta$ The closer $\tilde{\lambda}$ is to λ_1, the faster is the convergence of preconditioned gradient method.

Proof. Let the iteration process converges to the first eigenpair. Then for sufficiently great k it is possible to take $\lambda_k \to \lambda_1$ and $\alpha_k = \alpha$. From (13) yields:

$$\begin{aligned}
&\mathbf{g}_{k+1} = \left[\mathbf{I} - 2\alpha\mathbf{B}^{-1}\left(\mathbf{K} - \lambda_1\mathbf{M} + \delta\mathbf{M} - \delta\mathbf{M}\right)\right]\mathbf{g}_k = \\
&= \left[\mathbf{I} - 2\alpha\mathbf{B}^{-1}\left(\mathbf{B} - \delta\mathbf{M}\right)\right]\mathbf{g}_k = \\
&= \left[\left(1 - 2\alpha\right)\mathbf{I} + 2\alpha\delta\mathbf{B}^{-1}\mathbf{M}\right]\mathbf{g}_k
\end{aligned} \tag{16}$$

Let us take $\alpha = 0.5$. Then

$$\mathbf{g}_{k+1} = \left(\delta\mathbf{B}^{-1}\mathbf{M}\right)\mathbf{g}_k \tag{17}$$

This expression

$$\mathbf{g}_k = \sum_{i=2}^{N} \gamma_i^k \mathbf{v}_i \tag{18}$$

is an eigenvector expansion:

$$\mathbf{B}^{-1}\mathbf{M}\mathbf{v}_i = s_i \mathbf{v}_i, \quad i = 2,...,N \tag{19}$$

where N is number of equations of (1). The subscript i starts from 2, because the rank of space for eigenvectors (19) is $N-1$. So, the term $i = 1$ is used for scaling of (18).

The eigenvector orthogonality leads to:

$$\gamma_i^{k+1} = \delta s_i \gamma_i^k, \quad i = 2,...,N \tag{20}$$

If $\mathbf{B} = \mathbf{K} - \tilde{\lambda}\mathbf{M}$, then expression (19) leads to this:

$$\mathbf{K}\mathbf{v}_i = \underbrace{\left(\tilde{\lambda} + \frac{1}{s_i}\right)}_{=\tilde{\lambda}_i}\mathbf{M}\mathbf{v}_i, \quad i = 2,...,N, \tag{21}$$

$$s_i = \frac{1}{\lambda_i - \tilde{\lambda}}, \quad i = 2,...,N \tag{22}$$

The expression (20), taking into account (22), gives this:

$$\gamma_i^k = \left(\frac{\delta}{\lambda_i - \tilde{\lambda}}\right)^k \gamma_i^0 = \left(\frac{\delta}{\lambda_i - \lambda_1 + \delta}\right)^k \gamma_i^0, \quad i = 2,...,N \tag{23}$$

Expression (23) sets up the following important properties:

- The closer δ is to zero (as far as the existence of the inverse preconditioning matrix $\mathbf{B}^{-1}$ allows), the faster components of the gradient vector descent.
- Let us take $\lambda_1 < \lambda_2 < ... < \lambda_N$. Then $\lambda_i - \lambda_1 + \delta \approx \lambda_i - \lambda_1$, because it is possible to neglect the small value δ comparing to $\lambda_i - \lambda_1$. Expression (23) then simplifies to

$$\gamma_i^k \approx \left(\frac{\delta}{\lambda_i - \lambda_1}\right)^k \gamma_i^0, \quad i = 2,...,N \tag{24}$$

So, the convergence of high modes (large numbers of i) is faster than that of low modes (small numbers of i).

The properties 1, 2 are proved for the first eigenmode. The orthogonalization against already extracted modes decreases the dimensionality of the problem $N - n$ where n is the number of obtained modes. So, it is possible to generalize the above stated consideration to the second mode, third and so on.

Proposition: The properties 1, 2 suggest us to search the efficient preconditioning in the form:

$$\mathbf{B} = \mathbf{B}_0 - \tilde{\lambda}\mathbf{M} \tag{25}$$

where $\mathbf{B}_0$ is the aggregation multilevel preconditioning [6], [7], [8], [9] and $\tilde{\lambda}$ is a properly chosen shift.

4. An algorithm of the modified conjugate gradient method (MPCG) with a shifted aggregation multilevel preconditioning

The conventional algorithm of the PCG method for eigenvalue problem solution is presented in (11). The introduction of shifts into the preconditioning leads to a modification of this algorithm, because each shift update changes the spectral properties of matrices $\mathbf{A}, \mathbf{C}$ and destroys accumulated conjugate directions.

The following modifications are introduced:

- The preconditioning is taken in the form of (25)
- The shift update $\tilde{\lambda} = \lambda_{\tilde{K}}$ ($\lambda_{\tilde{K}}$ is an eigenvalue approximation at $\tilde{K}$ iteration step) is performed if the convergence has not been achieved during the prescribed number of iterations $\tilde{K}$. The iterations start from the beginning, and the initial eigenvector approximation is taken as the current eigenvector — see p.9 of the algorithm presented below. We avoid updating the shift too often, because we want to make use of the advantages of the conjugate gradient method. On the other hand, our aim is to accelerate the convergence by correcting the shift if a slow convergence occurs. The choice of $\tilde{K}$ depends on a problem and the efficiency of preconditioning. It is usually taken as $\tilde{K} = 20 \div 100$ for an aggregation multilevel

preconditioning. It turns out that after each correction the shift takes the value of the current eigenvalue approximation.

- The shifted preconditioning is not evaluated directly. An implicit iterative algorithm, presented at the end of this paragraph, is applied instead.

The algorithm of the MPCG method

1. Loop over all required modes $i = 1,2,...,n$
2. $\tilde{\lambda} = 0, \quad \hat{\mathbf{x}}_0 = 1$
3. Initialize the start vector
   - Orthogonalize against previously obtained modes $\varphi_1, \varphi_2, ..., \varphi_{i-1}$:
   - Normalize the respective mass matrix:
   - Approximate the eigenvalue: $\lambda_0 = \dfrac{(\mathbf{K}\mathbf{x}_0, \mathbf{x}_0)}{(\mathbf{M}\mathbf{x}_0, \mathbf{x}_0)}$
   - Residual vector:
     $\mathbf{r}_0 = -\mathbf{g}_0 = -(\mathbf{K}\mathbf{x}_0 - \lambda_0 \mathbf{M}\mathbf{x}_0), \quad \mathbf{B}\mathbf{z}_0 = \mathbf{r}_0 \Rightarrow \mathbf{z}_0$
   - New conjugate search direction:
     $\mathbf{p}_0 = -2(\mathbf{M}\mathbf{x}_0, \mathbf{x}_0)\mathbf{z}_0$
   - Orthogonalize $\mathbf{p}_0$ against previously defined modes $\varphi_1, \varphi_2, ..., \varphi_{i-1}$
4. PCG iterative loop: $k = 0,1,..., \tilde{K}$
5. Line search:
   $\hat{\mathbf{x}}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$, where
   $\alpha_k \to \min\{R(\hat{\mathbf{x}}_{k+1})\} = \min\left\{\dfrac{(\mathbf{K}_\sigma \hat{\mathbf{x}}_{k+1}, \hat{\mathbf{x}}_{k+1})}{(\mathbf{M}\hat{\mathbf{x}}_{k+1}, \hat{\mathbf{x}}_{k+1})}\right\}$
6. Normalize: $\mathbf{x}_{k+1} = \dfrac{\hat{\mathbf{x}}_{k+1}}{\sqrt{(\hat{\mathbf{x}}_{k+1}, \mathbf{M}\hat{\mathbf{x}}_{k+1})}}$
7. Approximate the eigenvalue: $\lambda_{k+1} = \dfrac{(\mathbf{K}\mathbf{x}_{k+1}, \mathbf{x}_{k+1})}{(\mathbf{M}\mathbf{x}_{k+1}, \mathbf{x}_{k+1})}$
8. Update the residual vector:
   $\mathbf{r}_{k+1} = -\mathbf{g}_{k+1} = -(\mathbf{K}\mathbf{x}_{k+1} - \lambda_{k+1}\mathbf{M}\mathbf{x}_{k+1}), \quad \mathbf{B}\mathbf{z}_{k+1} = \mathbf{r}_{k+1} \Rightarrow \mathbf{z}_{k+1}$
9. Check the convergence:
   $if\left(\|\mathbf{r}_{k+1}\| / \|\lambda_{k+1}\mathbf{M}\mathbf{x}_{k+1}\| \le tol\right)$
   Store $\varphi_i = \mathbf{x}_{k+1}; \quad \lambda_i = \lambda_{k+1}$ as a converged eigenpair,
   $i = i + 1$ and go to 1
   $else \quad if\left(k > \tilde{K}\right)$
   Modify: $\tilde{\lambda} = \lambda_{k+1}; \quad \mathbf{x}_0 = \mathbf{x}_{k+1}$, $k = 0$ and go to 3
   $else \quad if\left(k \le \tilde{K}\right)$ go to 10
10. Search for a new conjugate direction
    $\mathbf{B}\mathbf{z}_{k+1} = \mathbf{r}_{k+1} \to \mathbf{z}_{k+1}$
    $\beta = \dfrac{2}{(\mathbf{M}\mathbf{x}_{k+1}, \mathbf{x}_{k+1})} \dfrac{(\mathbf{z}_{k+1}, \mathbf{K}_\sigma \mathbf{p}_k)}{(\mathbf{p}_{k+1}, \mathbf{K}_\sigma \mathbf{p}_k)}$
    $\hat{\mathbf{p}}_{k+1} = \beta \mathbf{p}_k - \dfrac{2\mathbf{z}_{k+1}}{(\mathbf{M}\mathbf{x}_{k+1}, \mathbf{x}_{k+1})}$
11. Orthogonalize $\hat{\mathbf{p}}_{k+1}$ against previously defined modes $\varphi_1, \varphi_2, ..., \varphi_{i-1}$
12. End of loop k
13. End of loop i

The MPCG algorithm presented here is a generalization of the PCG one from [11] (pp. 132-133) by introducing the shift into the preconditioning and a proper shift correction during the iterations. If $\tilde{K} \to \infty$, the MPCG method tends to a usual PCG one.

The implicit solution scheme is applied to avoid the direct re-evaluation of the shifted preconditioning:

$$\left(\mathbf{B}_0 - \tilde{\lambda}\mathbf{M}\right)\mathbf{z}_{k+1} = \mathbf{r}_{k+1} \tag{26}$$

- $\mathbf{B}_0 \mathbf{z}^0_{k+1} = \mathbf{r}^0_{k+1} \Rightarrow \mathbf{z}^0_{k+1}$
- $\Delta = 0; \quad s = 1; \quad \mathbf{z}_{k+1} = \mathbf{z}^0_{k+1}$
- loop $s = 1,2,3,...$
  - $\mathbf{b} = \tilde{\lambda}\mathbf{M}\mathbf{z}_{k+1}$
  - $\mathbf{B}_0 \Delta = \mathbf{b} \Rightarrow \Delta$
  - $\mathbf{z}_{k+1} = \mathbf{z}^0_{k+1} + \Delta$
- End of loop s

Here Δ is a correction vector. Numerous numerical tests show that a single iteration loop ($s = 1$) is enough to ensure a good solution of (26) at the preconditioning stage.

5. Numerical results

5.1. Example 1

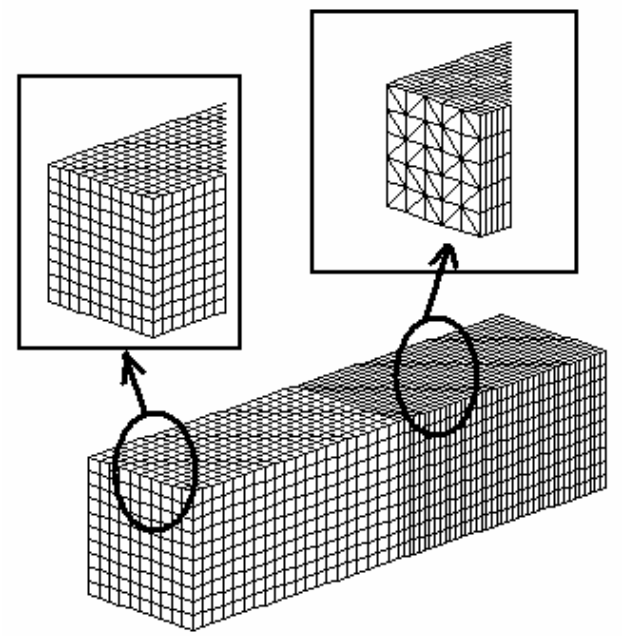


Fig.2 A spatial bar.

Table 1: Effect of shifts on the number of iterations and computation time (Example 1).

Mode #	Eigenvalue λ	Number of iterations for shift correction over 50 iterations ($\widetilde{K} = 50$)	Number of iterations for shift correction over 100 iterations ($\widetilde{K} = 100$)	Number of iterations without shifts ($\widetilde{K} \to \infty$)
1	9.556e+04	63	102	Convergence has not been achieved even for the first mode. 6 700 iterations were done, more than 12 h of computation time was spent
2	9.556e+04	19	20	
3	1.284e+06	26	26	
4	2.449e+06	24	24	
5	2.449e+06	27	27	
Total number of iterations		159	199	—
Total solution time, s		1 457	1 674	—

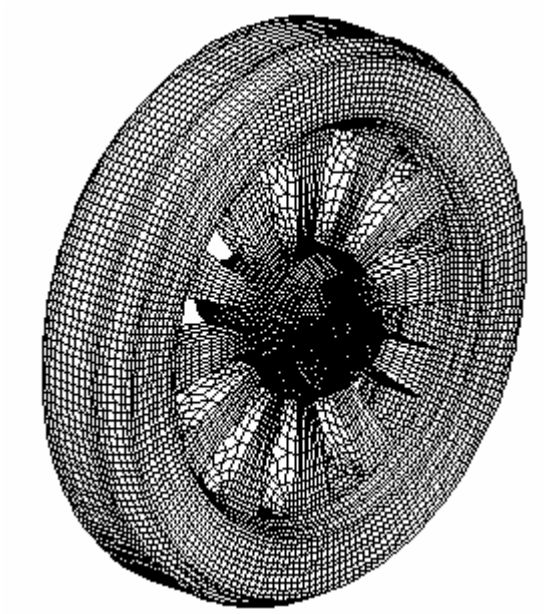


Fig.3 A finite element model of a wheel

A steel prismatic bar 4.12×1.39×1.39 m consisting of two parts with different meshes (Fig.2) has been considered. Its finite element model contains 58 621 nodes, 90 000 finite elements and 172 980 equations. The scheme presented in Fig. 2 has much coarser mesh than in reality because the real mesh is too dense to be shown clearly. The right part consists of brick volumetric finite elements and the left part includes 6-noded ones. 5 eigenpairs have been extracted. The precision is taken as $\|\mathbf{Kx} - \lambda\mathbf{Mx}\|_2 / \|\lambda\mathbf{Mx}\|_2 \le tol = 1.0 \times 10^{-4}$. A Pentium-III PC (CPU Intel 866 MHz, 256 MB RAM) has been used. Three aggregation levels (3 870 coarsest level equations for direct solution) and 4 inner iterations have been involved.

Table 2: Number of iterations and computation time for AMIS (aggregation multilevel MPCG method) and ICCF (incomplete Cholesky PCG method) methods

Method	Total number of iterations	Solution time, s
AMIS MPCG	159	1 457
ICCF PCG	8 281	14 554

The number of iterations and time of computation are presented in Table 1. It is clearly seen that the conventional PCG method has a lock of convergence at the first mode. The use of MPCG method allows us to overcome this difficulty due to properly selected shift values.

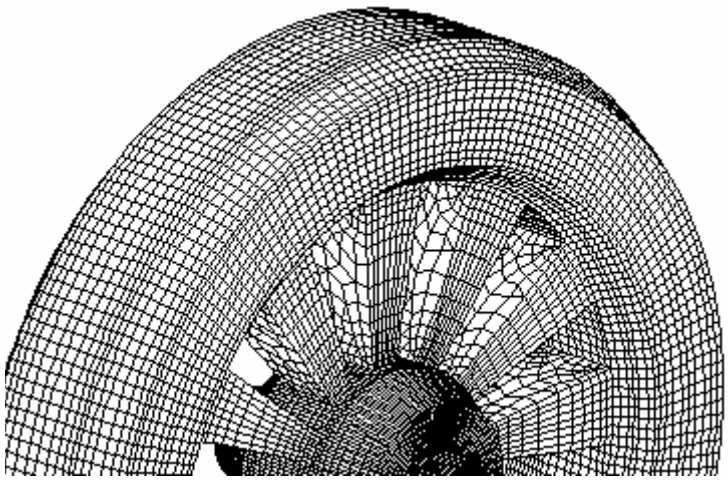


Fig. 4 A fragment of the mesh

The advantage of the aggregation multilevel modified preconditioned conjugate gradient method proposed here comparing to the conventional incomplete Cholesky preconditioned conjugate gradient one [9] is illustrated by data of Table 2.

5.2. Example 2

A finite element model of a wheel contains 96 451 nodes, 94 032 finite elements and 285 894 equations. (Fig.3, Fig.4)

Six eigenpairs have been extracted, $tol = 1.0 \times 10^{-4}$. Three aggregation levels (4 488 coarse level equations) and 4 inner iterations have been used. A Pentium-III computer (CPU Intel 1000 MHz, 512 MB RAM) has been applied.

The conventional PCG method leads to a lock of the convergence for the second mode. The computation was interrupted after over 9 hours. The MPCG method proposed here overcomes this difficulty (see Fig. 5, Table 3).

This problem is very hard for a direct solver. The stiffness matrix has 2 794 MB nonzero entries (the nested dissection

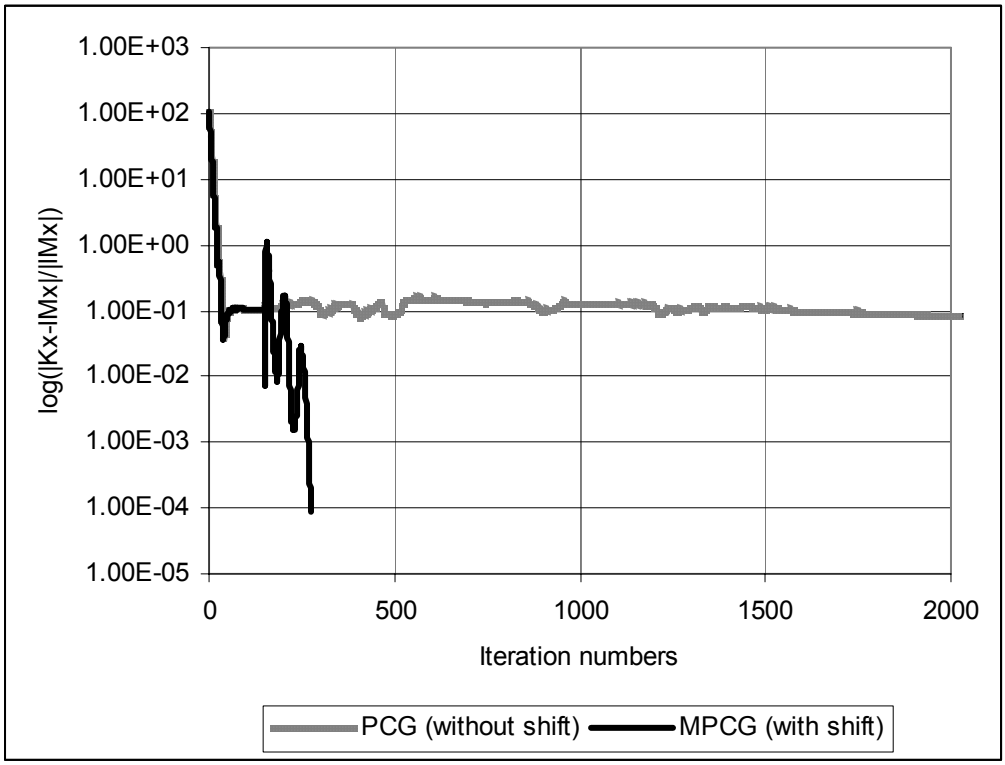


Fig.5 Second mode – the usual PCG method has a lock of convergence, the MPCG method overcomes this problem

Table 3: Effect of shifts on the number of iterations and computation time

Mode #	Eigenvalue λ	Number of iterations ($\widetilde{K} = 150$)	Number of iterations ($\widetilde{K} \to \infty$)
1	6.535e+003	88	88
2	1.191e+004	278	2039
3	1.191e+004	67	—
4	2.245e+005	75	—
5	2.807e+005	282	—
6	2.807e+005	58	—
Total number of iterations		848	—
Total computation time		3 h 43 m 23 s	>> 9 h

reordering method has been used as preferable to the minimum degree algorithm [8]). The stiffness matrix factoring time of the multi-frontal solver exceeds 9 hours which is more than the total solution time of MPCG method (3 h 43 m 23 s).

5.3. Example 3

A soil-structure interaction problem (Fig.6) is under consideration. The finite element model contains 104 048 nodes, 111 269 finite elements and 319 133 equations. A non-uniform mesh on the soil is denser in the building foundation area.

The sparse direct multi-frontal solver (the nested dissection reordering method) [8] requires about 1292 MB RAM only for allocation of the maximal front (18 403 equations) which is much more than the available storage on our computer (PC Pentium-III, CPU Intel-1000 MHz, 512 MB RAM). So the solution of this problem by methods based on factoring of the stiffness matrix (Lanczos method, subspace iteration and so on) requires a computer with larger RAM.

The aggregation multilevel MPCG method presented here allows us to successfully solve this problem on our available PC. The aggregation model consists of 3 aggregation levels (27 140 equations for direct solution), 4 inner iterations are used. The required size of RAM for the allocation of the compressed stiffness matrix (without any zero entries) and the factored part of the preconditioning is 285 MB. The tolerance 1.0×10^{-3} is taken, 10 eigenpairs are extracted, $\widetilde{K} = 60$

A comparison of the computational effort with that of the conventional PCG method (incomplete Cholesky factorization preconditioning) is presented in Table 4.

Table 4: Computational effort of different iterative methods Example 3

Method	Total number of iterations	Solution time, s
AMIS MPCG	667	3 h 26 m 05 s
ICCF PCG	6 204	5 h 20 m 06 s

Table 5 illustrates the soil-structure interaction effect for low vibration modes. The second model is the same building constrained on the foundation level.

The natural vibration modes for a soil-structure system differ essentially from those of a constrained structure. The compliance of the soil area adjoining the structure's foundation leads to reduced natural vibration frequencies comparing to those of the constrained structure. It can be clearly seen for first and third modes.

Even the general behavior of modes is essentially different in these two models (see Fig. 7).

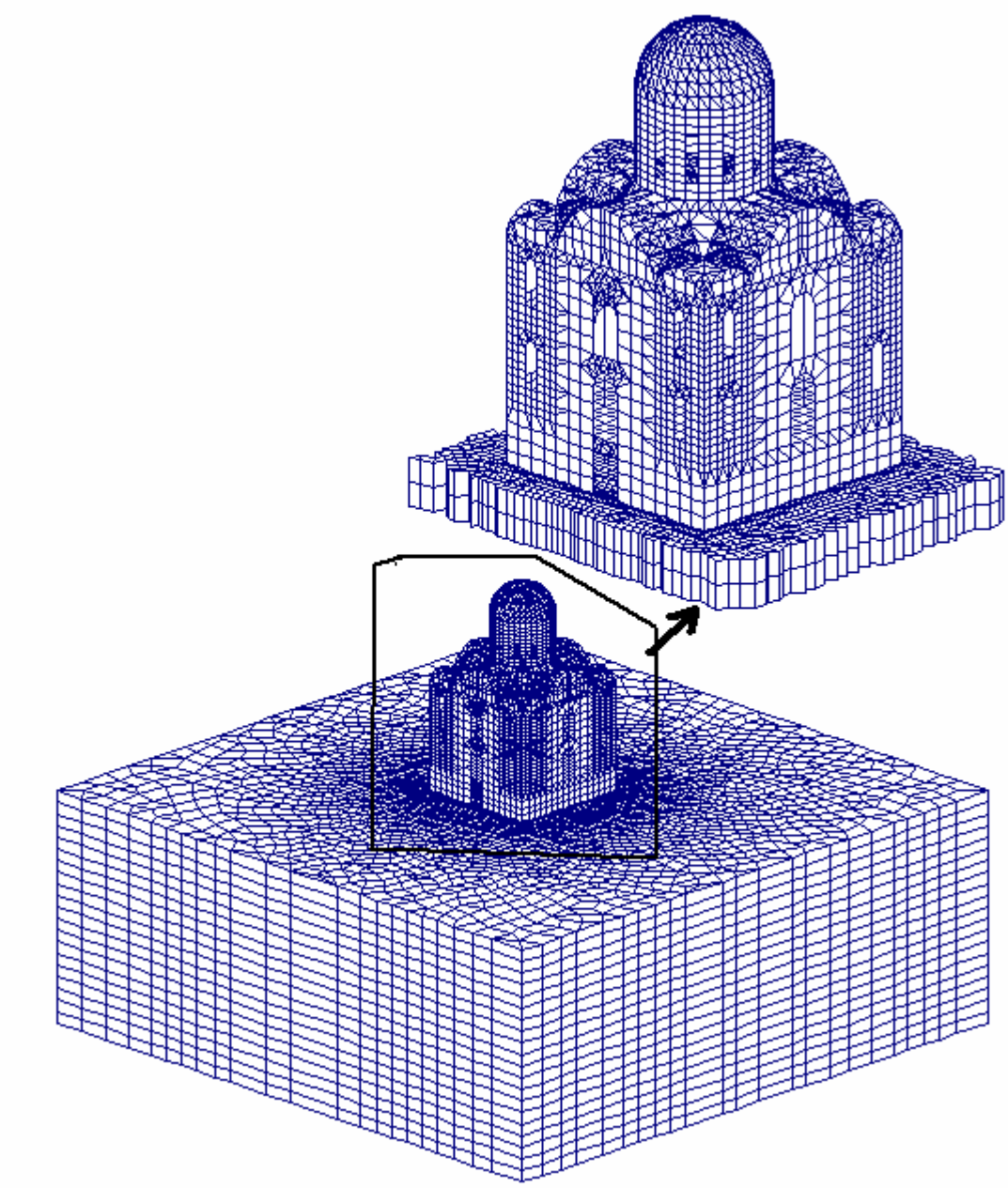


Fig.6 An example of a soil-structure interaction

6. Conclusion

The modified preconditioned conjugate gradient method presented here, one based on an aggregation multilevel preconditioning with a shift acceleration, is a powerful tool for natural vibration analysis of large-scale finite-element structural models.

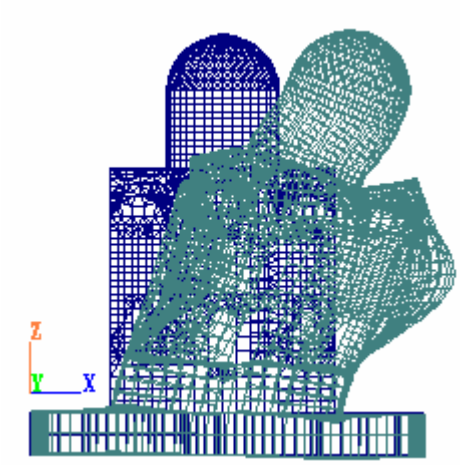
The aggregation approach allows us to analyze a variety of types of structures: bars, plates, shells, solids and any combinations of those. Any arbitrary types of finite elements from finite element libraries of commercial or research software are available.

The implementation of the shift results is an essential improvement of the preconditioning, acceleration of the convergence and avoidance of the convergence lock which often occurs with the sequential preconditioning conjugate gradient method. The aggregation approach together with the modified PCG algorithm proposed here, ensuring the proper choice of the shift value, allows us to keep the advantages of the preconditioned conjugate gradient method, accelerate the convergence and create a robust iterative technique.

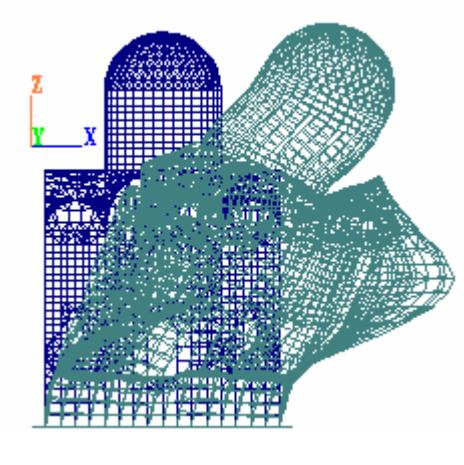
The numerical examples above prove the stable convergence and efficiency of the proposed technique. Soil-structure interaction problems are especially hard for methods based on a stiffness matrix factoring. The soil brick is a part of the stiffness matrix poorly optimized by reordering. Therefore

such methods require a lot of RAM and time-intensive computations. A robust iterative method, such as the aggregation multilevel MPCG technique proposed here, allows one to analyze this problem using a conventional PC.

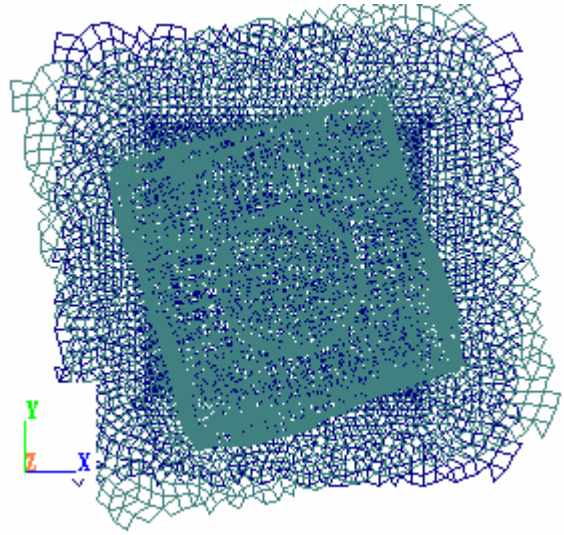
Table 6: Comparison of natural vibration modes in soil-structure interaction problems



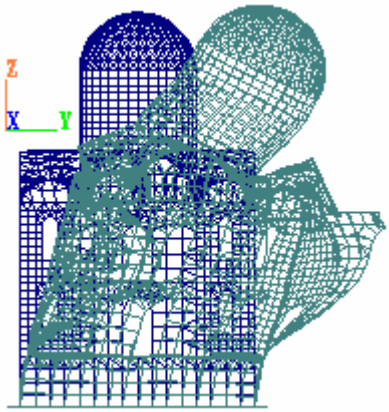
Mode 1 f = 2.86 Hz



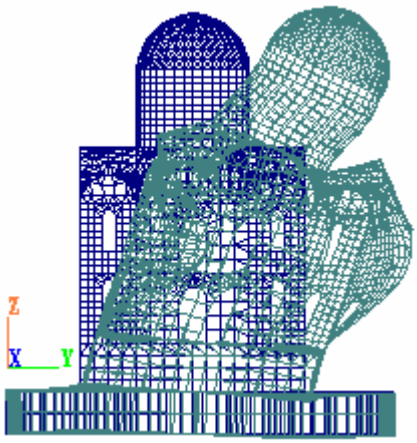
Mode 1 f = 3.18 Hz



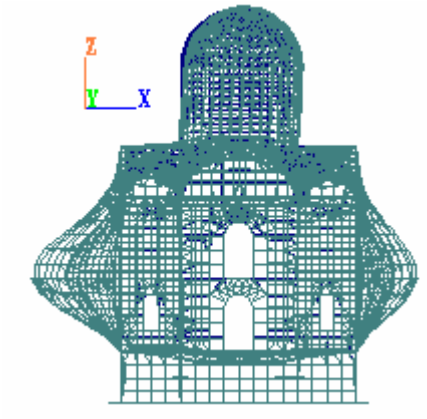
Mode 2 f = 3.02 Hz



Mode 2 f = 3.55 Hz



Mode 3 f = 3.07 Hz



Mode 3 f = 4.79 Hz

Fig 7 First three modes. A soil-structure system — on the left; a constrained structure — on the right

The Robot Millennium software (www.robobat.com) has been used to create the models of Examples 1, 2. The SCAD software (www.scadgroup.com) has been used to create the model of Example 3.

References

[1]. Bulgakov, V.E., Belyi, M.E., Mathisen, K.M., Multilevel aggregation method for solving large-scale generalized eigenvalue problems in structural dynamics, *Int. j. Numer. Methods Eng*, 40**,** pp. 453 – 471, 1997.

[2]. Bulgakov, V.E., Kuhn, G., High-performance multilevel iterative aggregation solver for large finite-element structural analysis problems, *Int. j. Numer. Methods Eng.,* 38**,** pp. 3529-3544, 1995.

[3]. Brandt, A., Multi-level adaptive solutions to boundary-value problems, *Mathematics of Computations*, 31, N 138, pp. 333-390, 1977.

[4]. Clough, R., Penzien, J., *Dynamics of structures*, McGraw-Hill, Inc., 1975.

[5]. Feng, Y.T., An integrated Davidson and multigrid solution approach for very large scale symmetric eigenvalue problems, *Comput. Meths. Appl. Mech. Eng*. 190, pp. 3543-3563 ,1999.

[6]. Fialko, S.Yu., The high-performance aggregation element-by-element iterative solver for the large-scale complex shell structural problems, *Archives of Civil Eng*., XLV, 2, pp. 193-207, 1999.

[7]. Fialko, S., High-performance aggregation element-by-element Ritz-gradient method for structure dynamic response analysis, *CAMES*, 7, pp. 537-550, 2000.

[8]. Fialko, S.Yu., High-performance iterative and sparse direct solvers in Robot software for static and dynamic analysis of large-scale structures. *Proceedings of the second European conference on computational mechanics, Cracow, Poland, June 26-29*, pp. 18, 2001.

[9]. Fialko, S. Aggregation Multilevel Iterative Solver for Analysis of Large-Scale Finite Element Problems of Structural Mechanics: Linear Statics and Natural Vibrations. *LNCS 2328*, p. 663 ff, http://link.springer.de/link/service/series/0558/tocs/t2328.htm

[10]. Hackbush, W., Trottenberg, U., *Multigrid Methods*, Springer-Verlag, Berlin, 1992.

[11]. Papadrakakis, M., *Solving large –scale problems in mechanics*, John Wiley & Sons Ltd., 1993.

[12]. Wilson, E.L., *Three dimensional dynamic analysis of structures*, Computers and Structures, Inc., Berkeley, California, USA, 1996.

[13]. Young Cho, Yook-Kong Yong, A multi-mesh, preconditioned conjugate gradient solver for eigenvalue problems in finite element models, *Computers & Structures*, 58, 3, pp. 575-583, 1996.