Computer Assisted Mechanics and Engineering Sciences, 7: 537–550, 2000.

High-performance aggregation element-by-element Ritz-gradient method for structure dynamic response analysis

Sergey Fialko
Kiev-190, ul. Chernjahovskogo 4 apt. 47, Ukraine, 252190,

(Received November 3, 1999)

The article presents the high-performance Ritz-gradient method for the finite element (FE) dynamic response analysis. It is based on the generation of the orthogonal system of the basis vectors. The gradient approach with two-level aggregation preconditioning on the base of element-by-element technique is applied to minimize the Rayleigh quotient for the preparation of each basis vector. It ensures the evolution of the regular basis vector toward the lowest eigenmode without aggregating and decomposing the large-scale stiffness matrix. Such method often happens to be more effective for dynamic response analysis, when compared to the classical modal superposition method, especially for seismic response analysis of the large-scale sparse eigenproblems. The proposed method allows one to apply arbitrary types of finite elements due to aggregation approach, and ensures fast problem solution without considerable exigencies concerning the disk storage space required, which is due to the use of EBE technique. This solver is implemented in commercial programs RobotV6 and Robot97 (software firm RoboBAT) for the seismic analysis of large-scale sparse problems and it is particularly effective when the consistent mass matrix is used.

1. Introduction

At present, the Lanczos and block subspace iteration methods are used as powerful tools in commercial programs where finite element analysis is implemented. They are applied to solve large-scale sparse eigenproblems, when it is necessary to derive a large number of eigenpairs [14, 17, 19]. Their application, however, is usually restricted by long computation time and huge disk storage requirements during Cholessky decomposition or Gauss elimination of the large-scale sparse stiffness matrix $\mathbf{K}$. This problem remains particularly acute when the consistent mass matrix is used. It forces one to turn to the iteration methods that do not require the above mentioned decomposition.

The papers [9, 10, 13–16] and others contributed greatly to the development of iterative methods for solving large-scale sparse finite-element eigenproblems. The preconditioned conjugate gradient method [10, 13–15] is usually applied to solve such problems. The multi-level (multi-grid) high-performance iterative methods [1–3, 8, 20] as well as the incomplete Cholessky factorization methods [10, 14, 15] have been developed intensively for the last years. The modified subspace iteration method [4] is of particularly great interest. It combines the idea of simultaneous subspace iterations with the iterative preconditioned gradient multi-level aggregation approach. All the above-mentioned approaches concentrate on the derivation of the low eigenmodes and eigenfrequencies.

This paper attempts to elaborate high-performance Ritz-gradient method that would allow us to derive Ritz vectors for the analysis of dynamic response of structures. The evaluation of exact (in the range of a given discrete model) eigenpairs is a very expensive procedure for large-scale sparse problems. Moreover, the classical modal superposition approach is still just as ineffective for some tasks of the seismic analysis (see [17–19]). Instead of deriving exact eigenpairs, the basis vector generation procedure is applied to evaluate the Ritz approximations. The evolution of each regular basis vector toward the lowest eigenvector is obtained by means of Rayleigh quotient minimization. The two-level preconditioned gradient method is used to derive good approximations of low Ritz

vectors to the corresponding eigenvectors. The orthogonalization procedure of the current basis vector against the previously defined ones is used to ensure its linear independence.

Detailed analysis shows that the Ritz vectors, derived by the application of the two-level preconditioned procedure, approximate better the global vibration modes than local ones. So, the proposed method tends to dump a part of local vibration modes. It is very important for the seismic analysis, because the local modes usually do not have an essential contribution to the seismic response due to small mass participation. But the presence of a large number of local modes into the low part of spectra requires huge computation effort, especially for certain kinds of bar and shell structures. The use of Ritz vectors leads to the implementation of a smaller number of base vectors which is the essential advantage of the proposed method.

The effectiveness of the preconditioned gradient method is achieved by defining appropriate properties of preconditioning [3, 8, 9, 11–14, 20]. The two-level preconditioned approach is applied here. The aggregate method [4–6] based on the EBE (element-by-element) technique is used to create a coarce level model. The main advantage of the aggregate approach consists in allowing one to use arbitrary types of finite elements. There is no necessity to elaborate specific high-order hierarchical finite elements, which is done in well-known fast iteration solvers [8, 14]. The usual finite element library can be applied both for the proposed method, and for the direct methods. The term "direct" denotes the well-known subspace iteration method, Lanczos method and others that use the decomposed stiffness matrix $\mathbf{K}$. It facilitates greatly the application of the proposed method to the existing FEM software. So, this method allows one to calculate bar and combined bar-continual structures as well – both as continual structures. This is very important for its application in commercial programs developed for analyzing and designing building structures.

The EBE technique envelops all stages of the proposed approach, namely: prolongation–restriction operations, the aggregation of coarse level matrix $\mathbf{K}_c$ and EBE preconditioning [11, 12] on inner smoothed iterations (see [3–6]). It ensures minimized disk space requirements and fast resolution of problems, due to respectively small size of data during input/output operations involving disk (secondary) memory.

2. Evolution of Ritz-gradient vectors

There is an eigenvalue problem

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

where $\mathbf{K}$, $\mathbf{M}$ are the stiffness and mass matrices, respectively, φ is the eigevector and λ is the eigenvalue. The procedure of evolution of the basis vector's set $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n$ toward the lowest eigenmode will be described. The preconditioned gradient approach is applied to minimize the Rayleigh quotient

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

where $k \in [0, n]$, k is the evolution step number; $n+1$ is the number of basis vectors that define the size of subspace *span* $\in (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n)$; $n+1 \ll N$ where N is the number degrees of freedom for the considered problem (1). It happens very often that the considered eigenvalue problem (1) is ill-conditioned. In such case, the evolution of the regular basis vector $\mathbf{x}_k$ toward the lowest eigenmode will be very slow. The preconditioning operator $\mathbf{B}$ is applied to improve such situation. The expression $\mathbf{B}\mathbf{z}_k = \mathbf{r}_k \Rightarrow \mathbf{z}_k$ means that the vector $\mathbf{z}_k$ is the resolution of the given equation set where $\mathbf{B}$ is preconditioning operator and $\mathbf{r}_k = \mathbf{K}\mathbf{x}_k - \lambda_k\mathbf{M}\mathbf{x}_k$ corresponds to the residual vector.

The base vectors satisfy the following conditions of orthogonality:

$$(\mathbf{M}\mathbf{x}_k, \mathbf{x}_p) = \left\{ \begin{array}{ll} 1, & k = p \\ 0, & k \neq p \end{array} \right\}, \qquad k, p = 0, 1, 2, \ldots, n. \tag{3}$$

The original large-scale eigenproblem (1) is reduced to the following subspace eigenproblem

$$\{k_{i,j}\}\vec{q} - \omega^2\{m_{i,j}\}\vec{q} = 0. \tag{4}$$

The matrices of subspace projection are defined as $\{k_{i,j}\} = (\mathbf{K}\mathbf{x}_i, \mathbf{x}_j)$ and $\{m_{i,j}\} = (\mathbf{M}\mathbf{x}_i, \mathbf{x}_j) = \mathbf{U}$ where $\mathbf{U}$ is a unit matrix.

The Ritz vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n+1}$ for derived base vectors $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n$ and the corresponding approximations of frequencies $\omega_1, \omega_2, \ldots, \omega_{n+1}$ are used for the superposition of the structural dynamic response.

The details of Ritz-gradient method are presented below.

Step 1. Start initialization

$$\tilde{\mathbf{x}}_0 = \begin{vmatrix} 1 \\ \vdots \\ 1 \end{vmatrix}, \qquad \mathbf{x}_0 = \frac{\tilde{\mathbf{x}}_0}{\sqrt{(\mathbf{M}\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_0)}}, \qquad \lambda_0 = \frac{(\mathbf{K}\mathbf{x}_0, \mathbf{x}_0)}{(\mathbf{M}\mathbf{x}_0, \mathbf{x}_0)},$$

$$\mathbf{r}_0 = \mathbf{K}\mathbf{x}_0 - \lambda_0\mathbf{M}\mathbf{x}_0, \qquad \mathbf{B}\mathbf{z}_0 = \mathbf{r}_0 \Rightarrow \mathbf{z}_0.$$

Step 2.

$$k = 0.$$

Step 3. Evolution of the regular basis vector

1. Search of the next basis vector to minimize the Rayleigh quotient into the $\mathbf{z}_k$ direction (see [10, 13–15])

$$a = (\mathbf{K}\mathbf{x}_k, \mathbf{z}_k), \quad c = (\mathbf{M}\mathbf{x}_k, \mathbf{z}_k), \quad m = (\mathbf{M}\mathbf{x}_k, \mathbf{x}_k),$$
$$b = (\mathbf{K}\mathbf{z}_k, \mathbf{z}_k), \quad d = (\mathbf{M}\mathbf{z}_k, \mathbf{z}_k), \quad n = (\mathbf{K}\mathbf{x}_k, \mathbf{x}_k),$$

$$\Delta = (nd - mb)^2 - 4(bc - ad)(ma - nc), \qquad \alpha_k = \frac{nd - mb + \sqrt{\Delta}}{2(bc - ad)} \tag{5}$$

$$\mathbf{x}^*_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{z}_k.$$

2. Orthogonalization against the previously derived basis vectors

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{x}^*_{k+1} - \sum_{i=0}^{k} \xi_i\mathbf{x}_i, \qquad \xi_i = (\mathbf{x}^*_{k+1}, \mathbf{M}\mathbf{x}_i).$$

3. Normalization of the current vector $(\mathbf{M}\mathbf{x}_{k+1}, \mathbf{x}_{k+1}) = 1$

$$\rho^2 = (\mathbf{M}\tilde{\mathbf{x}}_{k+1}, \tilde{\mathbf{x}}_{k+1}), \qquad \mathbf{x}_{k+1} = \frac{1}{\rho}\tilde{\mathbf{x}}_{k+1}.$$

Step 4. Compute $k + 1$ row of the stiffness and mass matrix projections to the $span\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{k+1}\}$

$$k_{k+1,j} = (\mathbf{K}\mathbf{x}_{k+1}, \mathbf{x}_j) \qquad m_{k+1,j} = \left\{ \begin{array}{ll} 1 & \text{for } k+1 = j \\ 0 & \text{for } k+1 \neq j \end{array} \right\}, \qquad j = 0, 1, \ldots, k+1.$$

Step 5. Compute the new search direction

$$\lambda_{k+1} = \frac{(\mathbf{K}\mathbf{x}_{k+1}, \mathbf{x}_{k+1})}{(\mathbf{M}\mathbf{x}_{k+1}, \mathbf{x}_{k+1})}, \qquad \mathbf{r}_{k+1} = \mathbf{K}\mathbf{x}_{k+1} - \lambda_{k+1}\mathbf{M}\mathbf{x}_{k+1}, \qquad \mathbf{B}\mathbf{z}_{k+1} = \mathbf{r}_{k+1} \Rightarrow \mathbf{z}_{k+1}.$$

Step 6.

if $(k+1 < n)$ *go to Step 3*
else perform Step 7.

Step 7. Solve the reduced eigenproblem

$$\{k_{ij}\}\mathbf{s} - \omega^2\{m_{ij}\}\mathbf{s} = \mathbf{0}\,, \qquad \mathbf{s} = \begin{vmatrix} s_0 \\ s_1 \\ \vdots \\ s_n \end{vmatrix}, \qquad i,j = 0,1,2,\ldots,n.$$

Step 8. Derive Ritz vectors

$$\mathbf{v}_i = \sum_{j=1}^{n+1} s_{i-1,j-1}\,\mathbf{x}_{j-1}\,, \qquad i = 1,2,\ldots,n+1.$$

The procedure of the evolution of the basis vectors $\mathbf{x}_k$, $k = 0,1,\ldots,n$, toward the lowest eigenmode is very close to the corresponding step of the preconditioned gradient iteration method for eigenproblem solution. It is noted in the numerous articles [1–3, 8, 11, 12, 20] that the convergence of the preconditioned iteration methods depends essentially on properties of preconditioned operator $\mathbf{B}$. This operator should be positive definite, it should allow an economic resolution of $\mathbf{B}\mathbf{z}_{k+1} = \mathbf{r}_{k+1}$ and it should satisfy to the condition number $\mathbf{C}(\mathbf{B}^{-1}\mathbf{K}) \mapsto 1$ as well as possible.

The last requirement in the case of Ritz-gradient method ensures a good approximation of the low part of eigenmodes.

3. Two-level aggregation element-by-element procedure

Let us introduce the following terms: fine level – it is the original finite-element model; coarse level – it is the reduced finite-element model, derived from the fine level by means of some restriction procedure. The subscripts f, c denote the fine level and coarse level correspondingly.

The main idea of the two-level approach consists in substituting the explicit resolution procedure $\mathbf{B}\mathbf{z}_{k+1} = \mathbf{r}_{k+1}$ with the following implicit one [3–6] (lower subscript $k+1$ will further on be omitted):

- Restriction of r vector to the coarse level, $r_f \mapsto \mathbf{r}_c$. This procedure consists in 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 (see [4–6]). The upper subscript ^{T} denotes the transposition.

- Resolution $\mathbf{K}_c\mathbf{z}_c = \mathbf{r}_c$ 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 one to implement the "direct" methods. Cholessky factorization method is used here).

- The prolongation $\mathbf{z}_c \mapsto \mathbf{z}_f$ from the coarse level to the fine level. This operation consists in 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 $\mathbf{z}_f$ after prolongation. The rapid-fluctuation residuals appear during the prolongation. The inner iteration procedure is applied to dump the residuals.

The coarse level model should predict low vibration modes. In this case, such two-level procedure of Ritz-gradient method leads to a good approximation of the low vibration modes, due to fast damping of the higher components of residuals. It has been shown [3] that the preconditioned gradient methods are characterized by fast convergence for the high-modal components of the resultant vector and by slow convergence for the low-modal components. So, such methods damp the high-modal components of the residual vector $\mathbf{r}$ very fast, but the low-modal components are reduced very slowly. The significance of the proposed two-level implicit preconditioning procedure consists in the fact that it predicts well the low-modal components of the resultant vector by means of the coarse level model, and it damps effectively the high-modal residuals by applying the inner iteration procedure during smoothing. The background of such theoretical positions is presented in [1–6, 20].

4. Approximation of the low vibration modes and connection with Lanczos method

Let us consider some important properties of the proposed Ritz-gradient method.

Theorem 1. *If the preconditioned operator* $\mathbf{B}$ *coincides with stiffness matrix* $\mathbf{K}$*, the proposed basis vector evolution procedure is still identical with Lanczos method.*

Proof. Let us rewrite (5) as follows,

$$\begin{aligned}\mathbf{x}_{k+1}^{*} &= \mathbf{x}_k + \alpha_k \mathbf{z}_k = \mathbf{x}_k + \alpha_k \mathbf{B}^{-1}\mathbf{r}_k \\ &= \mathbf{x}_k + \alpha_k \mathbf{B}^{-1}(\mathbf{K}\mathbf{x}_k - \lambda_k \mathbf{M}\mathbf{x}_k) = (1+\alpha_k)\mathbf{x}_k - \alpha_k \lambda_k \mathbf{K}^{-1}\mathbf{M}\mathbf{x}_k\,. \end{aligned} \tag{6}$$

The expression (6) with $\alpha_k = -1$ takes the form

$$\mathbf{x}_{k+1}^{*} = \lambda_k \mathbf{K}^{-1}\mathbf{M}\mathbf{x}_k\,. \tag{7}$$

The following orthogonalization procedure against the previously obtained basis vectors $\mathbf{x}_k$ and the normalization (see steps 2, 3 in the above presented algorithm) yields the set of basis vectors that are equal to the corresponding Lanczos vectors [14, 17, 18].

Theorem 2. *The regular basis vector* $\mathbf{x}_k$ *is obtained by prolongation and smoothing procedures applied to the corresponding coarse level vector* $\mathbf{x}_k^c$*, which is a result of current step of inverse iteration on coarse level.*

Proof. Let us present the fine level basis vector $\mathbf{x}_k$ as a result of prolongation and smoothing of the corresponding coarse level vector $\mathbf{x}_k^c$ (the subscript c denotes the coarse level),

$$\mathbf{x}_k = \mathit{Smooth}\{\mathbf{Q}\mathbf{x}_k^c\}\,. \tag{8}$$

The $\mathit{Smooth}\{\ldots\}$ denotes the smoothing operator. Let us apply (8) to (5),

$$\begin{aligned}\mathit{Smooth}\{\mathbf{Q}\mathbf{x}_{k+1}^c\} &= \mathit{Smooth}\{\mathbf{Q}\mathbf{x}_k^c\} + \alpha_k \mathit{Smooth}\{\mathbf{Q}\mathbf{z}_k^c\} = \mathit{Smooth}\{\mathbf{Q}(\mathbf{x}_k^c + \alpha_k \mathbf{z}_k^c)\} \\ &= \mathit{Smooth}\{\mathbf{Q}(\mathbf{x}_k^c + \alpha_k \mathbf{K}_c^{-1}\mathbf{r}_k^c)\} \\ &= \mathit{Smooth}\{\mathbf{Q}[\mathbf{x}_k^c + \alpha_k \mathbf{K}_c^{-1}(\mathbf{K}_c\mathbf{x}_k^c - \lambda_k^c \mathbf{M}_c\mathbf{x}_k^c)]\}\,. \end{aligned} \tag{9}$$

Let us note that $\alpha_k = -1$ yields

$$\mathbf{x}_{k+1}^c = \mathbf{x}_k^c + \alpha_k \mathbf{K}_c^{-1}(\mathbf{K}_c\mathbf{x}_k^c - \lambda_k^c \mathbf{M}_c\mathbf{x}_k^c) = \lambda_k^c \mathbf{K}_c^{-1}\mathbf{M}_c\mathbf{x}_k^c\,. \tag{10}$$

The expression (10) is a step of the inverse iteration [7] on coarse level. So, it follows from (9),

$$\mathbf{x}_{k+1}^{*} = \mathit{Smooth}\{\mathbf{Q}\mathbf{x}_{k+1}^c\} = \mathit{Smooth}\{\mathbf{Q}(\lambda_k^c \mathbf{K}_c^{-1}\mathbf{M}_c\mathbf{x}_k^c)\}\,. \tag{11}$$

The expression (11) sets up a connection between the inverse iteration step on coarse level and the regular basis vector $\mathbf{x}_{k+1}$ on fine level.

These theorems allow one to make the following conclusions that underline important properties of the basis vector set:

Conclusion 1. The difference between Lanczos method and proposed basis vectors evolution approach consists in the use of the prolongation and smoothing of the results at a step of the inverse iteration on coarse level instead of using it at a step of the inverse iteration on the fine level.

Conclusion 2. As the coarse level model approximates well the low vibration modes, the basis vectors $\mathbf{x}_k$ are close to the corresponding Lanczos vectors on fine level. In a limit case when $\mathbf{B} = \mathbf{K}$, basis vectors $\mathbf{x}_k$ are still exactly the Lanczos vectors on fine level.

5. AGGREGATION APPROACH AND ELEMENT-BY-ELEMENT TECHNIQUE

The main idea of aggregation method [4–6] is applied to derive the coarse level model. Details are presented in above mentioned articles. In this section, attention will focus on the application of the EBE technique to the aggregation of the coarse level matrix $\mathbf{K}_c$ and evaluation of the EBE restriction and prolongation operators. It is a very important moment, because it ensures fast preparation of $\mathbf{K}_c$ without any storage to disk of the large-scale sparse stiffness matrix $\mathbf{K}$. The matrix $\mathbf{K}_c$ is presented as

$$\mathbf{K}_c = \mathbf{Q}^T \mathbf{K} \mathbf{Q} \,. \tag{12}$$

The aggregation approach consists in introducing additional connections (rigid links) to decrease the number of degrees of freedom in a given computation model. The coarse level model is created in such a way. So, the original finite-element model (fine level) is transformed into a mechanical system (coarse level), consisting in rigid aggregates, coupled by the elastic connections. Rigid aggregates are the rigid bodies appearing due to imposed rigid links. All nodes of the finite-element model should be combined in the rigid aggregates. (It is possible to consider a single node as the limit case of the minimal rigid aggregate). It is not admissible for any node to be included in more than one aggregate.

A direct evaluation of the expression (12) is a very expensive way of obtaining the matrix $\mathbf{K}_c$. The EBE procedure,

$$\mathbf{K}_c = \sum_{e=1}^{N_e} \mathbf{T}_e^T \mathbf{K}_e \mathbf{T}_e \,, \tag{13}$$

is applied here to accelerate the computations. Here: N_e – the number of finite elements in a finite-element model, and $\mathbf{T}_e$ – the transformation matrix established due to additional rigid links. The restriction and prolongation operators ($\mathbf{Q}^T$ and $\mathbf{Q}$, respectively) are not evaluated directly, but implicit EBE procedures are applied during $\mathbf{r}_f \mapsto \mathbf{r}_c$ and $\mathbf{z}_c \mapsto \mathbf{z}_f$, respectively. This allows one to reduce the computation time and disk storage requirements for such operations.

The following expressions are generally correct, but we shell illustrate the transformation (4) on the particular example to simplify the understanding of this problem.

Let us consider the application of transformation (13) to the example of the 4-noded shell element. Let us assume that $i \in sAggr$, $j \in jAggr$, and $p, q \in iAggr$ (Fig. 1). Here, i, j, p, q – nodes of the considered shell element, and $iAggr$, $jAggr$, $sAggr$ – rigid aggregates.

The following is the equilibrium equation for the fine level model of the current element e,

$$\mathbf{K}_e \mathbf{q}_e = \mathbf{r}_e \,, \tag{14}$$

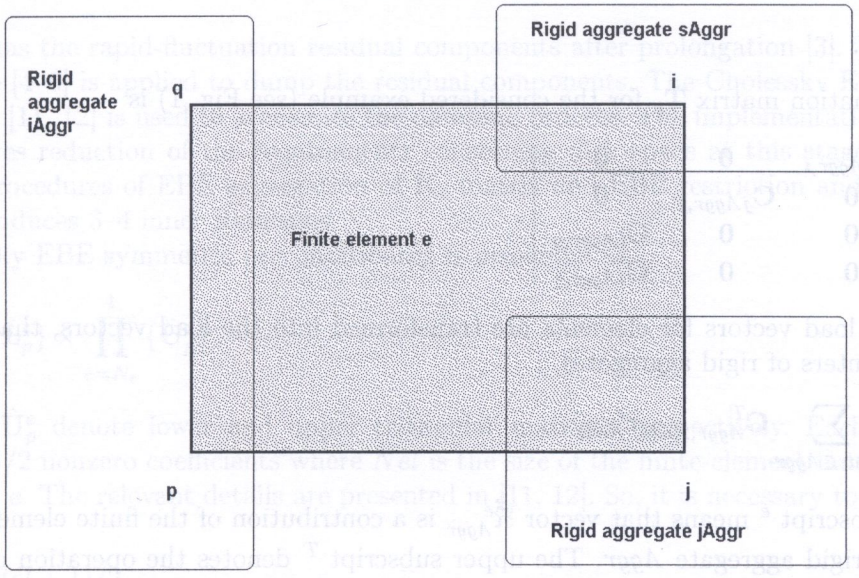

Fig. 1. Aggregation process for 4-noded shell element

where $\mathbf{K}_e$ – element stiffness matrix; $\mathbf{q}_e$ – the element nodal displacement vector; $\mathbf{r}_e$ – the element nodal load vector. The following are its values for the considered example,

$$\mathbf{K}_e = \begin{vmatrix} \mathbf{K}_{ii} & \mathbf{K}_{ij} & \mathbf{K}_{ip} & \mathbf{K}_{iq} \\ \mathbf{K}_{ji} & \mathbf{K}_{jj} & \mathbf{K}_{jp} & \mathbf{K}_{jq} \\ \mathbf{K}_{pi} & \mathbf{K}_{pj} & \mathbf{K}_{pp} & \mathbf{K}_{pq} \\ \mathbf{K}_{qi} & \mathbf{K}_{qj} & \mathbf{K}_{qp} & \mathbf{K}_{qq} \end{vmatrix}_e , \quad \mathbf{q}_e = \begin{vmatrix} \mathbf{u}_i \\ \mathbf{u}_j \\ \mathbf{u}_p \\ \mathbf{u}_q \end{vmatrix}_e , \quad \mathbf{r}_e = \begin{vmatrix} \mathbf{r}_i \\ \mathbf{r}_j \\ \mathbf{r}_p \\ \mathbf{r}_q \end{vmatrix}_e . \tag{15}$$

The block sub-matrices $\mathbf{K}_{ii}, \mathbf{K}_{ij}, \ldots, \mathbf{K}_{qq}$ contain stiffness coefficients for corresponding nodes.

The aggregation procedure at the level of elements leads to the following formula,

$$\mathbf{T}_e^T \mathbf{K}_e \mathbf{T}_e \tilde{\mathbf{q}}_e = \mathbf{T}_e^T \mathbf{r}_e \,. \tag{16}$$

The nodal displacement vector $\mathbf{u}_{Node}$ should be expressed through the displacement vector of the gravity center $\mathbf{y}_{Aggr}$ of the rigid aggregate $Aggr$ in the following manner,

$$\mathbf{u}_{Node} = \mathbf{C}_{Aggr,Node}\, \mathbf{y}_{Aggr} \,. \tag{17}$$

The values $Node$ and $Aggr$ denote a node number and the corresponding rigid aggregate number. It is evident that $Node \in Aggr$.

The matrix $\mathbf{C}_{Aggr,Node}$ should be evaluated as

$$\mathbf{C}_{Aggr,Node} = \begin{vmatrix} 1 & 0 & 0 & 0 & Z_{Node} - Z_{Aggr} & Y_{Aggr} - Y_{Node} \\ 0 & 1 & 0 & Z_{Aggr} - Z_{Node} & 0 & X_{Node} - X_{Aggr} \\ 0 & 0 & 1 & Y_{Node} - Y_{Aggr} & X_{Aggr} - X_{Node} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix} \tag{18}$$

where X_{Aggr}, Y_{Aggr}, Z_{Aggr} – the gravity center coordinates of the $Aggr$ rigid aggregate; X_{Node}, Y_{Node}, Z_{Node} – the coordinates of the node $Node \in Aggr$. The vector of the gravity center displacements of all rigid aggregates for finite element e is expressed by

$$\tilde{\mathbf{q}}_e = \begin{vmatrix} \mathbf{y}_{sAggr} \\ \mathbf{y}_{jAggr} \\ \mathbf{y}_{iAggr} \end{vmatrix} \tag{19}$$

and

$$\mathbf{q}_e = \mathbf{T}_e \tilde{\mathbf{q}}_e \,. \tag{20}$$

The transformation matrix $\mathbf{T}_e$ for the considered example (see Fig. 1) is

$$\mathbf{T}_e = \begin{vmatrix} \mathbf{C}_{sAggr,i} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{jAggr,j} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_{iAggr,p} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_{iAggr,q} \end{vmatrix} . \tag{21}$$

The nodal load vectors for elements are transformed into the load vectors, that are applied to the gravity centers of rigid aggregates,

$$\vec{\Re}^e_{Aggr} = \sum_{Node \in Aggr} \mathbf{C}^T_{Aggr,Node} \mathbf{r}^e_{Node} \,. \tag{22}$$

The upper subscript ^{e} means that vector $\vec{\Re}^e_{Aggr}$ is a contribution of the finite element e to the load vector of the rigid aggregate $Aggr$. The upper subscript ^{T} denotes the operation of transposition. So, the expression (16) should be presented in the following form,

$$\tilde{\mathbf{K}}_e \tilde{\mathbf{q}}_e = \tilde{\mathbf{r}}_e \,, \tag{23}$$

where

$$\tilde{\mathbf{K}}_e = \mathbf{T}^T_e \mathbf{K}_e \mathbf{T}_e \,; \qquad \tilde{\mathbf{r}}_e = \mathbf{T}^T_e \mathbf{r}_e = \begin{vmatrix} \vec{\Re}^e_{sAggr} \\ \vec{\Re}^e_{jAggr} \\ \vec{\Re}^e_{iAggr} \end{vmatrix} . \tag{24}$$

The coarse level matrix $\mathbf{K}_c$ should be aggregated by the EBE procedure

$$\mathbf{K}_c = \sum_{e=1}^{N_e} \tilde{\mathbf{K}}_e \,.$$

The EBE restriction procedure is evaluated as

$$\mathbf{r}_c = \sum_{e=1}^{N_e} \sum_{e \in Aggr} \vec{\Re}^e_{Aggr} = \sum_{Aggr=1}^{Naggr} \vec{\Re}_{Aggr} \,. \tag{25}$$

The symbol $e \in Aggr$ means that this sum is extended to such finite elements whose nodes are included into $Aggr$ aggregate. The number of rigid aggregates is denoted by $Naggr$. The vector $\vec{\Re}_{Aggr}$ is the load vector applied to the gravity center of the rigid aggregate $Aggr$,

$$\vec{\Re}_{Aggr} = \sum_{e=1}^{N_e} \vec{\Re}^e_{Aggr} \,.$$

The EBE prolongation procedure is evaluated as

$$\mathbf{z}_f = \sum_{Node=1}^{Nnodes} \mathbf{z}^f_{Node} = \sum_{Node=1}^{Nnodes} \mathbf{C}_{Aggr,Node} \, \mathbf{z}^c_{Aggr} \,. \tag{26}$$

The symbol $\mathbf{z}^f_{Node}$ denotes the displacement vector on the fine level for such components that are related to the node $Node$; symbol $\mathbf{z}^c_{Aggr}$ – the displacement vector of the gravity center of the rigid aggregate $Aggr$ and $Nnodes$ is the number of nodes of the finite-element model. The sums of (25), (26) imply the aggregation process.

6. Inner iteration procedure and element-by-element technique

The vector $\mathbf{z}_f$ has the rapid-fluctuation residual components after prolongation [3]. The inner iteration procedure [4–6] is applied to dump the residual components. The Cholessky EBE symmetric preconditioning [11, 12] is used to accelerate the damping process. The implementation of the EBE technique ensures reduction of the requirements concerning disk space at this stage as well as at the stage the procedures of EBE aggregation of $\mathbf{K}_c$ matrix and EBE restriction and prolongation. This usually produces 3–4 inner iterations.

The Cholessky EBE symmetric preconditioning is presented as

$$\tilde{\mathbf{B}}_{CH} = \prod_{e=1}^{N_e} \{\hat{\mathbf{L}}_p^e\} \times \prod_{e=N_e}^{1} \{\hat{\mathbf{U}}_p^e\}, \tag{27}$$

where $\hat{\mathbf{L}}_p^e$ and $\hat{\mathbf{U}}_p^e$ denote lower and upper triangular matrices, respectively. Each of them has $Nel \times (Nel+1)/2$ nonzero coefficients where Nel is the size of the finite-element matrix $\mathbf{K}_e$ for the current element e. The relevant details are presented in [11, 12]. So, it is necessary to store only the

$$\sum_{e=1}^{N_e} Nel \times (Nel+1)/2$$

coefficients (the matrix $\hat{\mathbf{U}}_p^e$ should be evaluated as a transposition of the corresponding matrix $\hat{\mathbf{L}}_p^e$), instead of storing the profile of the large-scale stiffness matrix $\mathbf{K}$. The fast resolution method is applied to solve the equation set

$$\tilde{\mathbf{B}}_{CH}\mathbf{y}_j = \mathbf{r}_j^* \tag{28}$$

where $\mathbf{y}_j$ – unknown vector, $\mathbf{r}_j^*$ – residual vector for j^{th} step of inner iteration. The fast resolution algorithm is presented by the following formulas,

$$\begin{aligned}
&\mathbf{y}_j^0 = \mathbf{r}_j^*, \\
&\hat{\mathbf{L}}_p^e \mathbf{y}_j^e = \mathbf{y}_j^{e-1} \Rightarrow y_j^e && e = 1, \ldots, N_e, \\
&\hat{\mathbf{U}}_p^e \mathbf{y}_j^{2N_e-e+1} = \mathbf{y}_j^{2N_e-e} \Rightarrow y_j^{2N_e-e+1}, && e = N_e, \ldots, 1, \\
&\mathbf{y}_j = \mathbf{y}_j^{2N_e}.
\end{aligned} \tag{29}$$

It is very easy to perform such procedure, because the matrices $\hat{\mathbf{L}}_p^e$ and $\hat{\mathbf{U}}_p^e$ are triangular with a small number of nonzero coefficients. Such matrices are not evaluated directly, but by means of references to the appropriate coefficients of the corresponding element matrices (see [11–12]).

7. Numerical results

7.1. Model example

As an example we will consider a thin square plate with consistent mass matrix which is clamped along one edge. We apply the 4-noded shell element and mesh 128×128 (number of equations Neq is 99072 and the bandwidth after optimization is 1242). The ten low frequencies obtained by both Lanczos and Ritz-gradient (PCG_Ritz) methods are presented in Table 1. It occurs that the residual norm $\|\mathbf{K}\boldsymbol{\varphi}_i - \omega_i^2\mathbf{M}\boldsymbol{\varphi}_i\|/\|\omega_i^2\mathbf{M}\boldsymbol{\varphi}_i\|$, $i = 1, 2, \ldots$, for results obtained by Lanczos method is usually smaller than 10e−7. So, it is possible to claim that Lanczos method produces the exact solution in the range of a given discrete model.

The frequencies obtained by means of PCG_Ritz methods are very close to corresponding frequencies obtained by means of Lanczos method. The PCG_Ritz method requires essentially less computation time and disk storage space (see Table 2) than Lanczos method.

Table 1. Natural vibration frequencies (Hz), obtained by Lanczos method and PCG_Ritz method

Mode number	Lanczos method	PCG_Ritz method	Error %
1	4.6865e+01	4.6966e+01	0.20
2	1.1282e+02	1.1298e+02	0.14
3	2.0833e+02	2.0902e+02	0.33
4	2.6751e+02	2.6781e+02	0.11
5	3.3584e+02	3.3622e+02	0.11
6	3.7696e+02	3.7841e+02	0.38
7	4.8816e+02	4.9048e+02	0.47
8	5.6343e+02	5.6960e+02	1.09
9	6.1785e+02	6,2137e+02	0.56
10	6.7662e+02	6.7757e+02	0.14

Table 2. Computation efforts for Lanczos method and PCG_Ritz method

Method	Lanczos method	PCG_Ritz method
Computation time, s	75 017	7 367
Disk storage requirements, MB	2 216	401

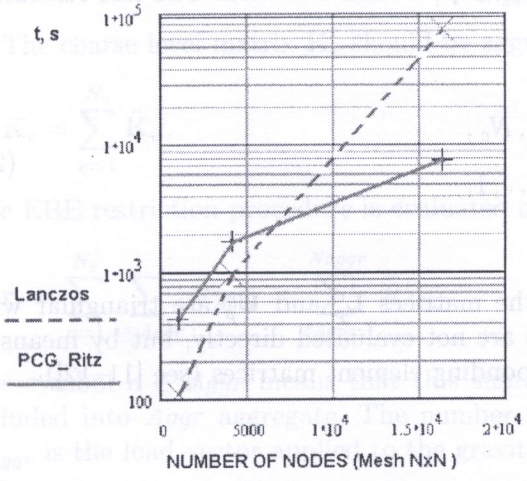

Fig. 2. Dependencies of computation time from mesh type

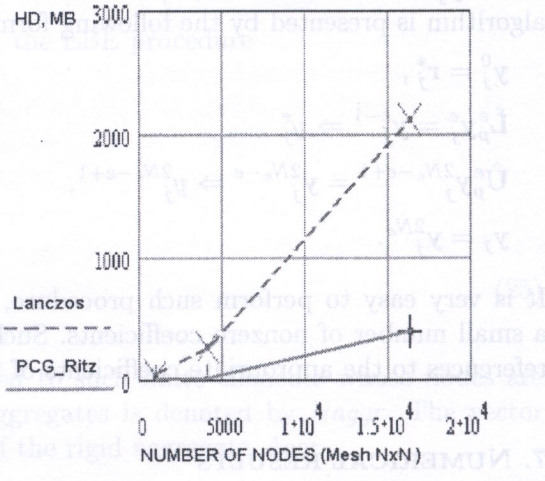

Fig. 3. Dependencies of disk storage requirements from mesh type

Table 3. Parameters of problems for given mesh types. *Neq* – number of equations, *Bandwidth* – bandwidth after optimization

Mesh	32×32	64×64	128×128
Number of nodes	1024	4096	16384
Neq/Bandwidth	6 336 / 324	24 960 / 630	99 072 / 1 242

The dependence of computation time (Fig. 2) and disk storage requirements (Fig. 3) on mesh type (see Table 3) indicates that computation efforts for "direct" methods catastrophically increase if the size of a problem increases. 40 Ritz-gradient vectors have been generated to derive 10 Ritz vectors of PCG_Ritz method for all considered problems. We used a computer Pentium 400 with 512 MB RAM.

7.2. The examples of real problems, taken from engineering practice

The following three quite different problems (see Figs. 4–6) are presented to illustrate the effectiveness of the proposed PCG_Ritz method. They are taken from practice of IT firm RoboBAT. The corresponding models consist of 3D bar and 3–4-noded shell elements. Parameters of problems and computational efforts are presented in Table 4 where Neq – number of equations, bandwidth – average bandwidth after optimization. Column 4 shows the number of required modes and the type of mass matrix (consistent or lumped). For both consistent and lumped mass matrices, the PCG_Ritz

Table 4. Parameters of problems and computation efforts

Problem	Neq	Band width	NModes / masses	Method	Time, s	Disk, MB	Computer
PJG203	34 266	990	25, consistent	Lanczos	32 420	598	Pentium PRO
				PCG_Ritz	3 912	85	
Cinema	41 118	1380	60, lumped	Lanczos	44 457	669	Pentium 120
				PCG_Ritz	24 597	165	
Telecom	83 574	2610	30, consistent	Lanczos	*Has not been solved*	4200	Pentium 400
				PCG_Ritz	36 473	364	Pentium 120

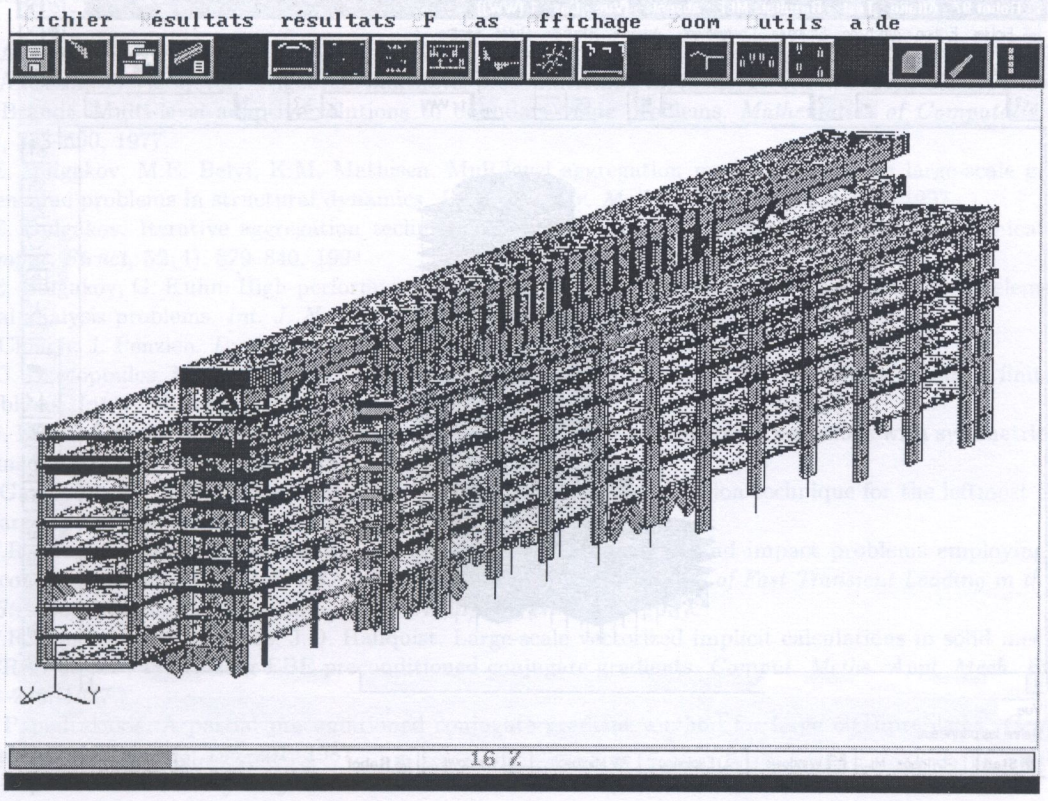

Fig. 4. Problem PJG203

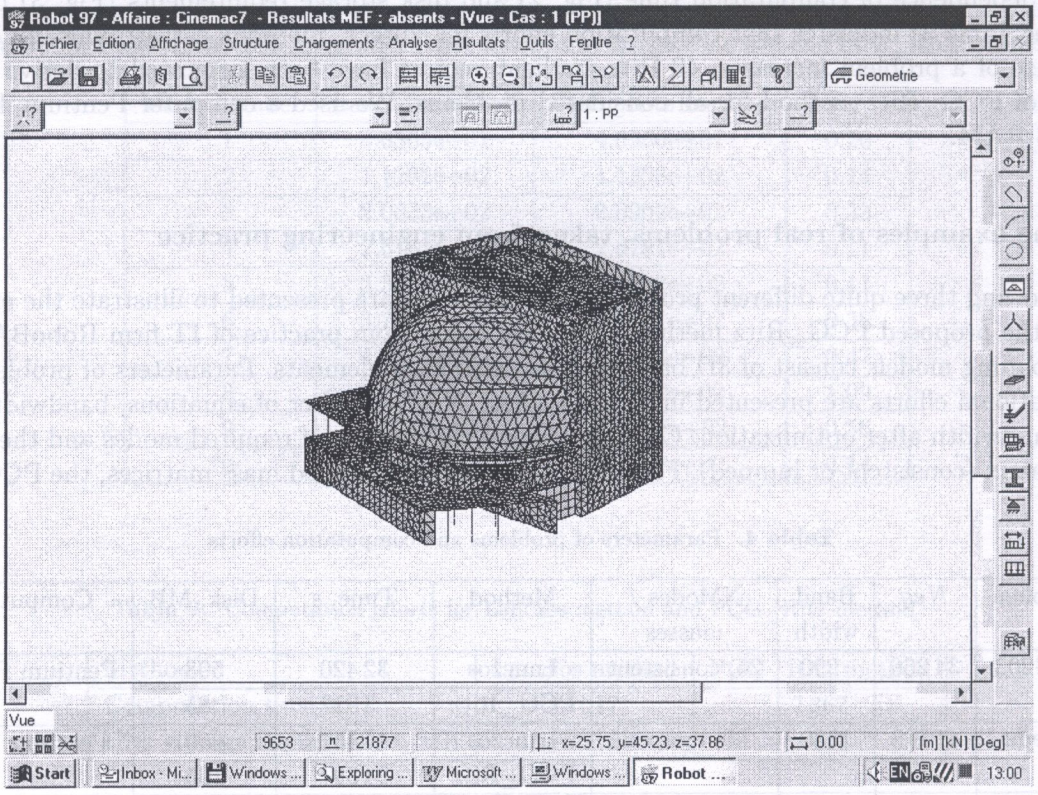

Fig. 5. Cinema building in the Near East (problem Cinema)

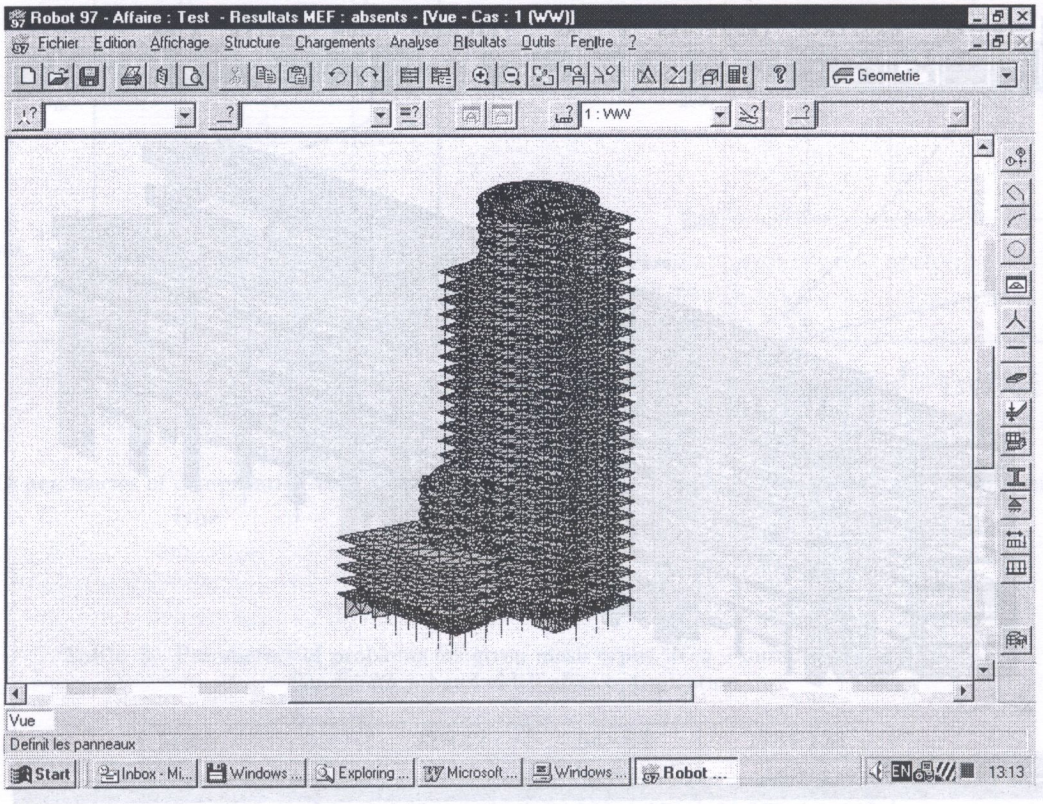

Fig. 6. Polish telecommunication multistory building (problem Telecom)

method turned out to be more effective than traditional Lanczos method. The last problem was impossible to solve – even on powerful computer Pentium 400 (512 MB RAM) – by Lanczos method, due to the requirement of huge disk storage space. However, the problem was solved successfully on an ordinary PC Pentium-120 (64 MB RAM) by means of the PCG_Ritz method.

8. Conclusion

The proposed Ritz-gradient method is a powerful tool for solving large-scale sparse dynamic problems. It allows one to obtain Ritz vectors that approximate the low eigenmodes. The quality of such approximation depends essentially on the properties of the preconditioning operator $\mathbf{B}$. Theorems 1, 2 and Conclusions 1, 2 establish the following general peculiarity of the proposed method. As the coarse level model approximates well the low vibration modes, the basis vectors $\mathbf{x}_k$ are close to the corresponding Lanczos vectors on fine level, and the Ritz vectors on fine level are good approximations of the corresponding eigenvectors.

The two-level aggregation approach ensures the preparation of a coarse level model that usually approximates well the low vibration modes. The application of EBE technique leads to the reduction of requirements on disk storage space and to fast problem resolution.

The above-mentioned properties of Ritz-gradient method allow one to reduce essentially the computation time and to maintain the precision that is sufficient for practical dynamic analysis of building structures.

Acknowledgments

This work was supported by informative firm RoboBAT, Kraków, Poland.

References

[1] O. Axelsson, P. Vassilevski. Algebraic multilevel preconditioning methods, I. *Num. Math.*, **56**: 157–177, 1989.

[2] O. Axelsson, P. Vassilevski. Algebraic multilevel preconditioning methods, II. *Num. Math.*, **57**: 1569–1590, 1990.

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

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

[5] V.E. Bulgakov. Iterative aggregation technique for large-scale finite element analysis of mechanical systems. *Comput. Struct*, **52**(4): 829–840, 1994.

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

[7] R. Clough, J. Penzien. *Dynamics of Structures*, McGraw–Hill, Inc., 1975.

[8] M.C. Dracopoulos, M.A. Criesfield. Coarse/fine mesh preconditioners for the iterative solution of finite element problems. *Int. J. Numer. Methods Eng.*, **38**: 3297–3313, 1995.

[9] D.J. Evans. The use of preconditioning in iterative methods for solving linear equations with symmetric positive-defined matrices. *J. Inst. Math. Appl.*, **4**: 295–314, 1976.

[10] G. Gambolati, G. Pini, F. Sartoretto. An improved iterative optimization technique for the leftmost eigenpairs of large symmetric matrices. *J. Comp. Phys.*, **74**: 41–60, 1988.

[11] T.J.R. Hughes, M. Ferencz. Implicit solution of large-scale contact and impact problems employing an EBE preconditioned iterative solver. *IMPACT 87 Int. Conference on Effects of Fast Transient Loading in the Context of Structural Mechanics, Lausanne, Switzerland, August 26–27, 1987.*

[12] T.J.R. Hughes, R.M. Ferencz, J.O. Hallquist. Large-scale vectorized implicit calculations in solid mechanics on a CRAY X-MP/48 utilizing EBE preconditioned conjugate gradients. *Comput. Meths. Appl. Mech. Engrg.*, **61**: 215–248, 1987.

[13] M. Papadrakakis. A partial preconditioned conjugate gradient method for large eigenproblems. *Comp. Meth. Appl. Mech. Eng.*, **62**: 195–207, 1987.

[14] M. Papadrakakis. *Solving Large-Scale Problems in Mechanics*, John Wiley and Sons Ltd., 1993.

[15] F. Sartoretto, G. Pini, G. Gambolati. Accelerated simultaneous iterations for large finite element eigenproblems. *J. Comp. Phys.*, **81**: 53–69, 1989.

[16] H.R. Schwarz, Two algorithms for treating $\mathbf{Ax} = \lambda\mathbf{Bx}$. *Comput. Meths. Appl. Mech. Eng.*, **12**: 181–199, 1977.
[17] E.L. Wilson. An eigensolution strategy for large systems. *Comput. Struct.*, **16**(1–4): 259–265, 1983.
[18] E.L. Wilson. A new method of dynamic analysis for linear and nonlinear systems. *Finite Elements in Analysis and Design*, **1**: 21–23, 1985.
[19] E.L. Wilson. *Three Dimensional Dynamic Analysis of Structures*, Computers and Structures, Inc., Berkeley, California, USA, 1996.
[20] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, **34**(4): 581–613, 1992.