Proceedings of the Federated Conference on Computer Science and Information Systems pp. 669–672
DOI: 10.15439/2016F88
ACSIS, Vol. 8. ISSN 2300-5963

Block Subspace Projection PCG Method for Solution of Natural Vibration Problem in Structural Analysis

Sergiy Fialko
Tadeusz Kościuszko Cracow University of Technology
ul. Warszawska 24 St., 31-155 Kraków, Poland
Email: sergiy.fialko@gmail.com

Filip Żegleń
Tadeusz Kościuszko Cracow University of Technology
ul. Warszawska 24 St., 31-155 Kraków, Poland
Email: filipzeglen@hotmail.com

***Abstract*—The block subspace projection preconditioned conjugate gradient method for analysis of natural vibration frequencies and modes applying to large problems of structural mechanics is proposed. It is oriented at the usage in finite element analysis software operated on multi-core desktop computers with restricted amount of core memory as an alternative approach to widespread block Lanczos method and subspace iteration method. We focused our attention on achievement of high computational stability and parallelization of proposed algorithm. The solution of real-life large problems confirms the reliability of proposed approach.**

I. Introduction

THE block Lanczos method as well as various versions of subspace iteration method widely is used for extraction of natural vibration frequencies and modes in modern engineering software applying to the problems of structural mechanics

$$\mathbf{K}\mathbf{v}_i - \lambda_i \mathbf{M}\mathbf{v}_i = 0 \ , \tag{1}$$

where **K** and **M** are the sparse symmetric stiffness and mass matrices arising when the finite element method is applied to problems of structural mechanics, $\{\lambda_i, \mathbf{v}_i\}$ – eigenpair for i-th mode, $i \in [1, n]$, $n << N$, n – number of required eigenmodes, N – dimension of problem (1).

The Lanczos method as well as subspace iteration (SI) approach produces the inverse iterations

$$\mathbf{K}\mathbf{v}_i^{k+1} = \mathbf{M}\mathbf{v}_i^k \ , \tag{2}$$

where $\mathbf{v}_i^k$ is an approximation of eigenvector $\mathbf{v}_i$ on iteration step k (k = 1, 2, … until converges). For large problems solved on desktop and laptop computers with restricted amount of core memory the lower triangular matrix **L** of factorized stiffness matrix $\mathbf{K} = \mathbf{L}\cdot\mathbf{D}\cdot\mathbf{L}^T$ is stored block-by-block on disk. Therefore, on each iteration step k mentioned above eigenvalue solvers must read twice the lower triangular matrix **L** from disk. Taking into account that size of matrix **L** for large design models (N = 3 000 000 – 6 000 000 equations) achieves 6 – 20 GB and more, performance of such eigenvalue solver drastically decreases.

Unlike mentioned Lanczos and SI approaches, the preconditioned conjugate gradient (PCG) method [8], [12] uses only RAM. However, for poorly conditioned problems, which are the most of problems of structural mechanics (see [7]), we have to construct an efficient preconditioning, because the conventional SSOR, symmetrical Gauss-Seidel, ICCG0 preconditioners result in unacceptable slow convergence. We found that the aggregation multilevel preconditioning [1], [2] and incomplete Cholesky factorization by value, realized in technique of sparse matrices [4], demonstrate a stable convergence for solution of linear equation sets (static analysis) as well as for extraction of natural vibrations frequencies and modes (modal analysis) for considered class of problems.

Achieving of stable convergence of the conjugate gradient method for solving the eigenvalue problem (1) is much more difficult than in solving systems of linear algebraic equations. Most likely, for this reason, in modern commercial FEA software mainly used eigenvalue solvers based on inverse matrix iteration (2) [9]. In present article, we propose the block subspace projection preconditioned conjugate gradient (BSPPCG) method for solution of problem (1).

II. Block Subspace Projection Preconditioned Conjugate Gradient Method

A. State of problem

To achieve the computational stability of PCG method at solution of poorly conditioned problems of structural mechanics, the aggregation multilevel preconditioning and shift technique have been used [1]. A little later, an aggregation multilevel preconditioning was replaced by an incomplete Cholesky factorization by value implemented in technique of sparse matrix [4].

Article [5] presents a block version of PCG method, where several vectors in block are iterated simultaneously. The Gram-Schmidt orthogonalization provides orthogonality of vectors in the block between themselves as well as their orthogonality to the eigenmodes, converged earlier. The shift technique is used for acceleration of convergence.

This work was not supported by any organization

A very interesting idea of local block PCG (LOBPCG) method was proposed in [10]. The approximations on the next iteration step are presented as:

$$\begin{cases} \mathbf{v}_i^{k+1} = \sum_{j=1}^{m} \alpha_j^k \mathbf{z}_j^k + \sum_{j=1}^{m} \tau_j^k \mathbf{v}_j^k + \sum_{j=1}^{m} \gamma_j^k \mathbf{p}_j^k \\ \mathbf{p}_i^{k+1} = \sum_{j=1}^{m} \alpha_j^k \mathbf{z}_j^k + \sum_{j=1}^{m} \gamma_j^k \mathbf{p}_j^k \end{cases}, \quad i \in [1, m], \quad (3)$$

where m is a dimension of subspace $span\{\mathbf{z}_1^k,\ldots, \mathbf{z}_m^k, \mathbf{v}_1^k,\ldots, \mathbf{v}_m^k, \mathbf{p}_1^k,\ldots, \mathbf{p}_m^k\}$, $\mathbf{z}_j^k = \mathbf{B}^{-1}\mathbf{r}_j^k$, $\mathbf{B}$ – preconditioning operator, $\mathbf{r}_j^k = \lambda_j^k \mathbf{M}\mathbf{v}_j^k - \mathbf{K}\mathbf{v}_j^k$ – residual vector, $\mathbf{p}_j^k$ – conjugate direction vector, $j = 1, \ldots, m$. The projection matrix $\mathbf{Q} = \{\mathbf{Z}\ \mathbf{V}\ \mathbf{P}\}$ has dimension $N \times 3{\cdot}m$ and consists of $N \times m$ submatrices $\mathbf{Z} = \{\mathbf{z}_1^k,\ldots, \mathbf{z}_m^k\}$, $\mathbf{V} = \{\mathbf{v}_1^k,\ldots, \mathbf{v}_m^k\}$ and $\mathbf{P} = \{\mathbf{p}_1^k,\ldots, \mathbf{p}_m^k\}$. In matrix form:

$$\begin{aligned} &\mathbf{V}^{k+1} = \mathbf{Q}\cdot\mathbf{q}, \quad \mathbf{q} = \{\mathbf{q}_1,\ldots,\mathbf{q}_m\}, \\ &\mathbf{q}_j = \left\{\boldsymbol{\alpha}_j^k\ \boldsymbol{\tau}_j^k\ \boldsymbol{\gamma}_j^k\right\}^T, \quad j \in [1, m] \end{aligned}, \quad (4)$$

where $\boldsymbol{\alpha}_j^k = \{\alpha_{1,j}^k, \ldots, \alpha_{m,j}^k\}$, $\boldsymbol{\tau}_j^k = \{\tau_{1,j}^k, \ldots, \tau_{m,j}^k\}$, $\boldsymbol{\gamma}_j^k = \{\gamma_{1,j}^k, \ldots, \gamma_{m,j}^k\}$ and iteration number k is omitted for matrices $\mathbf{Q}$, $\mathbf{q}$. Subscript j denotes a number of eigenmode in expansion (6).

Let us substitute (7) in (1) and multiple at left by $\mathbf{Q}^T$:

$$\mathbf{k}\mathbf{q} - \mathbf{m}\mathbf{q}\boldsymbol{\Lambda} = 0, \quad (5)$$

where $\mathbf{k} = \mathbf{Q}^T\mathbf{K}\mathbf{Q}$, $\mathbf{m} = \mathbf{Q}^T\mathbf{M}\mathbf{Q}$ and $\boldsymbol{\Lambda}$ is a diagonal matrix with approximations of eigenvalues on iteration step k. The approximation of eigenmodes $\mathbf{v}^{k+1}$ on the next iteration step follows from (4) after substitution of $\mathbf{q}$, which is obtained from solution of reduced eigenproblem (5). The conjugate direction vector is derived as

$$\mathbf{P}^{k+1} = \mathbf{Q}'^T\mathbf{q}', \quad (6)$$

where $\mathbf{Q}' = \{\mathbf{Z}\ \mathbf{P}\}$ and $\mathbf{q}' = \{\boldsymbol{\alpha}^k\ \boldsymbol{\gamma}^k\}^T$. Then, the Rayleigh quotient is used for approximation of eigenvalues on iteration step k+1 and evaluation of residual vectors $\mathbf{r}_j^k$ is produced after it.

The proposed approach in the form of [10] is little suitable to analysis of real-life problems of structural mechanics, since the columns in matrix $\mathbf{Q}$ become the linearly dependent as soon as the first eigenpair begins to converge. The authors of current article obtained the computational instability even for simple test problem – simply supported beam. The clear explanation of this fact is in [11]: the second expression in (3) is a linear combination between vectors $\mathbf{z}_j^k$ and $\mathbf{p}_j^k$. Therefore, the first expression in (3) is possible to rewrite as

$$\mathbf{v}_i^{k+1} = \sum_{j=1}^{m} \alpha_j^k \mathbf{z}_j^k + \sum_{j=1}^{m} \tau_j^k \mathbf{v}_j^k + \sum_{j=1}^{m} \beta_j^k \mathbf{v}_j^{k-1}, \quad i \in [1, m]. \quad (7)$$

For instance, let vector $\mathbf{v}_1$ begins to converge. Then, basis vectors $\mathbf{v}_1^k$ and $\mathbf{v}_1^{k-1}$ are almost linearly dependent, because $\mathbf{v}_1^{k-1}$, $\mathbf{v}_1^k$ as well as $\mathbf{v}_1^{k+1}$ tends to the same eigenvector $\mathbf{v}_1$ – an exact solution. In [11] for stabilization of computational process was suggested to remove the vectors $\mathbf{z}_j^k$, $\mathbf{v}_j^k$, $\mathbf{p}_j^k$ of subspace $span\{\mathbf{z}_1^k,\ldots, \mathbf{z}_m^k, \mathbf{v}_1^k,\ldots, \mathbf{v}_m^k, \mathbf{p}_1^k,\ldots, \mathbf{p}_m^k\}$, as soon as the corresponding vector $\mathbf{v}_j^k$ converges.

The drawbacks of such approach are:

- For some problems of structural mechanics it turns out that the iterative process is falling apart even before the error

$$err_i = \left\|\mathbf{r}_i^k\right\|_2 \Big/ \left[\lambda_i^k \left\|\left(\mathbf{v}_i^k\right)^T \mathbf{M}\mathbf{v}_i^k\right\|_2\right] \quad (8)$$

is still enough small ($err_j \le tol$) to recognize the convergence of vector $\mathbf{v}_j^k$.

- The dimension of subspace m must be not less than the number of required eigepairs n ($m \ge n$). Such an approach results in enormous computational efforts if it is required a large number of eigenpairs (n = 100 – 200 and more). In addition, it is required a significant amount of core memory for allocation of matrix $\mathbf{Q}$.

The proposed in given article eigenvalue block subspace projection PCG (BSPPCG) solver uses the idea (3) of LOBPCG, but possesses the high computational stability and requires essentially less computational efforts when large number of eigenpairs is required. That allows us to recommend this solver for use in FEA software intended for analysis of problems of structural mechanics on widespread desktop and laptop computers as well as on shared memory workstations.

B. Algorithm of BSPPCG method

1. Set $m \in [8, 32]$, $m\%np = 0$; prepare linearly independent start vectors $\mathbf{V}^0=\{\mathbf{v}_1^0,\ldots,\mathbf{v}_m^0\}$, $\mathbf{P}^0=\{\mathbf{p}_1^0,\ldots,\mathbf{p}_m^0\}$, $nconvmodes = 0$;
2. **do** $k = 1, 2, \ldots$, **until** $nconvmodes < n$.
3. **parallel loop for** $j = 1, \ldots, m$
   $(\mathbf{v}_j^k)^T{\cdot}\mathbf{M}{\cdot}\mathbf{v}_j^k = \mathbf{I}$ (normalization procedure)
   $\lambda_j^k = [(\mathbf{v}_j^k)^T\ \mathbf{K}\ \mathbf{v}_j^k]/[(\mathbf{v}_j^k)^T\ \mathbf{M}\ \mathbf{v}_j^k]$
   $\mathbf{r}_j^k = \lambda_j^k\mathbf{M}\mathbf{v}_j^k - \mathbf{K}\mathbf{v}_j^k$
   $\mathbf{B}\ \mathbf{z}_j^k = \mathbf{r}_j^k \rightarrow \mathbf{z}_j^k$
   end of parallel loop for
4. **do** j=1, m
   if($err_j \le tol$)
   $nconvmodes$++;
   put $\{\lambda_j, \mathbf{v}_j\}$ as final results,
   prepare new start vectors $\mathbf{v}_j^k$ and $\mathbf{p}_j^k$,
   orthogonalize $\mathbf{v}_j^k$ against converged modes
   and put to blocks $\mathbf{V}^k$, $\mathbf{P}^k$.
   $\lambda_j^k = [(\mathbf{v}_j^k)^T\ \mathbf{K}\ \mathbf{v}_j^k]/[\ [(\mathbf{v}_j^k)^T\ \mathbf{M}\ \mathbf{v}_j^k]$
   $\mathbf{r}_j^k = \lambda_j^k\mathbf{M}\mathbf{v}_j^k - \mathbf{K}\mathbf{v}_j^k$
   $\mathbf{B}\ \mathbf{z}_j^k = \mathbf{r}_j^k \rightarrow \mathbf{z}_j^k$, put $\mathbf{z}_j^k$ to $\mathbf{Z}^k$.

end if
end do
5. **paralel loop for** $s = 1, 3m$
loop for $p = s, 3m$ (s, p – columns of matrix $\mathbf{Q}$)
$\mathbf{m}_{sp} = \mathbf{Q}_p{}^T\mathbf{M}\mathbf{Q}_s$ – evaluation of matrix $\mathbf{m}$
end loop for
end of parallel loop for
6. **if**(Chol($\mathbf{m}$): $\mathbf{m} = \mathbf{l}\cdot\mathbf{l}^T$)
paralel loop for $s = 1, 3m$
loop for $p = s, 3m$ (s, p – columns of matrix $\mathbf{Q}$)
$\mathbf{k}_{sp} = \mathbf{Q}_p{}^T\mathbf{K}\mathbf{Q}_s$ – evaluation of matrix $\mathbf{k}$
end loop for
end of parallel loop for
else
Gram-Schmidt orthogonalization of all columns in $\mathbf{Q}$ and normalization $\mathbf{Q}^T\mathbf{M}\mathbf{Q} = \mathbf{I}$
paralel loop for $s = 1, 3m$
loop for $p = s, 3m$ (s, p – columns of matrix $\mathbf{Q}$)
$\mathbf{m}_{sp} = \mathbf{Q}_p{}^T\mathbf{M}\mathbf{Q}_s$ – evaluation of matrix $\mathbf{m}$
$\mathbf{k}_{sp} = \mathbf{Q}_p{}^T\mathbf{K}\mathbf{Q}_s$ – evaluation of matrix $\mathbf{k}$
end loop for
end of parallel loop for
Chol($\mathbf{m}$): $\mathbf{m} = \mathbf{l}\cdot\mathbf{l}^T$
end if
7. solve reduced eigenproblem $\mathbf{kq} - \mathbf{mq}\Lambda = 0$
8. obtain $\mathbf{V}^{k+1}$ and $\mathbf{P}^{k+1}$ using (4), (6).
9. parallel orthogonalization of $\mathbf{V}^{k+1}$ and $\mathbf{P}^{k+1}$ against converged eigenmodes.
end do

Algorithm 1. BSPPCG method

We accept the fixed dimension of subspace m, which is multiple to available number of threads np ($m\%np = 0$) for achievement a load balance between threads (point 1). Then, we prepare linearly independent start vectors $\mathbf{V}^0$ and set $\mathbf{p}_j^0 = \emptyset$, $j \in [1, m]$, where $\emptyset$ is a zero vector, and number of converged modes set to sero: $nconvmodes = 0$.

The iteration loop **do** k =1, 2,… runs until $nconvmodes < n$, where n is the number of required modes (point 2).

In parallel loop (point 3) for each mode j we produce the normalization of $\mathbf{v}_j^k$, obtain the current approximation of eigenvalue λ_j^k, residual vector $\mathbf{r}_j^k$ and vector $\mathbf{z}_j^k$ from solution of linear equation set arising when preconditioning operator $\mathbf{B}$ is introduced for acceleration of convergence. On first iteration, the projection matrix $\mathbf{Q}$ contains only submatrices $\mathbf{Z}$ and $\mathbf{V}$ because submatrix $\mathbf{P}$ is zero. On all subsequent iterations, $\mathbf{Q}$ comprises $\mathbf{Z}$, $\mathbf{V}$ and $\mathbf{P}$ submatrices.

Iteration loop **do** j=1, m (point 4) checks of convergence. If convergence of j-th mode is achieved, we store the eigenpair $\{\lambda_j , \mathbf{v}_j\}$ to structure of data containing the final results, increment $nconvmodes$ and prepare the new linearly independent start vectors in addresses of vectors $\mathbf{v}_j^k$ and $\mathbf{p}_j^k$. In each starting vector $\mathbf{p}_j^k$ we put only one element equal to unit, all remaining elements are zero. Due to such an action, all new vectors $\mathbf{p}_j^k$ are linearly independent. All remaining elements of each vector $\mathbf{p}_j^k$ are zero. In such a way, we avoid the linear dependency between columns of projection matrix $\mathbf{Q}$. Then, we orthogonalize the new starting vectors $\mathbf{v}_j^k$ against converged modes and compute the λ_j^k, $\mathbf{r}_j^k$ and $\mathbf{z}_j^k$ corresponding to new starting vectors $\mathbf{v}_j^k$. Vectors $\mathbf{z}_j^k$, corresponding to new starting vectors, replace columns j in submatrix $\mathbf{Z}$, corresponding to converged vectors on current iteration step k.

The reduced matrix $\mathbf{m}$ is evaluated in **parallel loop for** $s = 1, 3m$. The sparse matrix $\mathbf{M}$ is multiplied by columns $\mathbf{Q}_s$ of matrix $\mathbf{Q}$ in parallel region: $\mathbf{w}_s = \mathbf{K}\cdot\mathbf{Q}_s$. The number of threads in team is np. In second loop (**loop for** $p = s, 3m$) we calculate the element $\mathbf{m}_{sp}$ as a dot product of column $\mathbf{Q}_p{}^T$ and previously obtained vector $\mathbf{w}_s$. Matrix $\mathbf{m}$ is symmetrical therefore p starts with s.

Chol($\mathbf{m}$) denotes the Cholesky factorization of matrix $\mathbf{m}$ and $\mathbf{l}$ is a lower triangular matrix. (point 6). We consider the matrix $\mathbf{m}$ as a weighted Gram matrix of subspace $span\{\mathbf{z}_1{}^k,\ldots,\mathbf{z}_m{}^k, \mathbf{v}_1{}^k,\ldots,\mathbf{v}_m{}^k, \mathbf{p}_1{}^k,\ldots,\mathbf{p}_m{}^k\}$. Therefore, if Cholesky factorization completes successfully, the columns of matrix $\mathbf{Q}$ are linearly independent and basis vectors are OK. Otherwise, if Cholesky factorization of $\mathbf{m}$ is failed, the columns of matrix $\mathbf{Q}$ are almost linearly dependent, and we produce the Gram-Schmidt orthogonalization of all columns in $\mathbf{Q}$ and normalization $\mathbf{Q}^T\mathbf{M}\mathbf{Q} = \mathbf{I}$. After this, we prepare reduced matrices $\mathbf{k}$, $\mathbf{m}$ using multithreaded parallelization and repeat Cholesky factorization of $\mathbf{m}$.

We apply procedures from LAPACK of Intel math kernel library (Intel MKL) [13] for solution of the generalized algebraic eigenproblem (5).

Submatrices $\mathbf{V}^{k+1}$ and $\mathbf{P}^{k+1}$ (point 8) is derived using (4), (6). We apply the multithreaded version of *dgemm* procedure from Intel MKL for multiplication of dense matrices.

The Gram-Schmidt orthogonalization provides orthogonality of vectors in the submatrices $\mathbf{V}^{k+1}$ and $\mathbf{P}^{k+1}$ to the converged eigenmodes (point 9). The columns in $\mathbf{V}^{k+1}$ as well as in $\mathbf{P}^{k+1}$ can be independently orthogonalized against converged eigenmodes, therefore, this algorithm can be easily parallelized. Also, orthogonalization, made in point 4, can be easily parallelized. Unlike these algorithms, the Gram-Schmidt orthogonalization procedure applied to all columns of matrix $\mathbf{Q}$ (point 6) has a strongly sequential nature and cannot be successfully parallelized.

We emphasize the fundamental differences between proposed method and LOBPCG.

1. BSPPCG method keeps constant the dimension of subspace m, which does not depend on number of required

eigenmodes. This allows us to reduce the amount of core memory and computing time and makes proposed approach applicable for solution of large problems on desktops and laptops.

2. As soon as the vectors converge, we immediately remove them from the block and replace with new starting vectors. In many cases, this allows us to keep a linear independence of the columns in matrix **Q**.

3. If in spite of everything, linear dependencies between base vectors still appears, we make the full reortogonalization of columns of the matrix **Q**.

4. We apply an efficient preconditioning for considered class of problems – incomplete Cholesky factorization by value developed in technique of sparse matrices [4].

III. Numerical Results

We consider example "stadium" taken from computational practice of SCAD Soft IT Company, developer of the SCAD FEA software, one of the most popular software used in the CIS countries for structural analysis and design, certified according to the regional norms.

We use computer with 16-core processor AMD Opteron 6276, 2.3/3.2 GHz, 64 GB DDR3 RAM, OS Windows Server 2008 R2 Enterprise SP1, 64 bit. The large amount of RAM allows us on application of incomplete Cholesky factorization for preparation of preconditioning with very small value of drop parameter $\psi = 10^{-16}$ [4] and keep all data in core memory. In addition, 16 processor cores provide the opportunity to explore the speed-up of method when we increase the number of cores. The tolerance is accepted as $tol = 10^{-3}$ – see (8).

The design model of stadium comprises 4 033 620 equations and consists of several types of finite elements: spatial frames, triangular and quadrilateral flat shell finite elements, elastic supports and rigid links One hundred eigenpairs are extracted (n = 100). The large number of almost multiple natural vibration frequencies occurs due to local vibration modes of bars in spatial trusses.

For accepted values of ψ, tol the number of iterations is 121 and number of reorthogonalizations, when Cholesky factoring of matrix **m** was failed, is 20. If there is at least one reorthogonalization, that means that LOBPCG method in version [10] would fail, since the columns of the matrix **Q**, which are basis vectors, are linearly dependent. Reorthogonalization of columns in matrix **Q** allows us to successfully continue a computation process. The shortest computational time is achieved on 16 threads.

Table I presents the total time of eigenvalue analysis of considered problem using proposed BSPPCG method, shifted block PCG (SBPCG) method [5] and shifted block Lanczos (SBLANC) method [3]. Method SBLANC solves this problem in core memory using PARFES [6], [7] – one from fastest for today sparse direct solvers on shared memory computers.

TABLE I
COMPARISON OF COMPUTATIONAL TIME FOR DIFFERENT METHODS. COMPUTER A, PROBLEM 1.

Method	Total time, s
BSPPCG	6 466
SBPCG	20 708
SBLANC (core mode)	6 228

The proposed BSPPCG demonstrates the solution time, which is slightly bigger than solution time of SBLANC method. The SBPCG method solves this problem considerably slower.

References

[1] S. Yu. Fialko, "Natural vibrations of complex bodies," *Int. Applied Mechanics,* vol. 40, no. 1, pp. 83 – 90, 2004, http://DOI: 10.1023/B:INAM.0000023814.13805.34.

[2] S. Fialko, "Aggregation Multilevel Iterative Solver for Analysis of Large-Scale Finite Element Problems of Structural Mechanics: Linear Statics and Natural Vibrations", in *PPAM 2001*, R. Wyrzykowski et al. (Eds.), *LNCS* 2328, Springer-Verlag Berlin Heidelberg, 2002, pp. 663–670, http://DOI: 10.1007/1-4020-5370-3_41.

[3] S. Yu. Fialko, E. Z. Kriksunov and V. S. Karpilovskyy, "A block Lanczos method with spectral transformations for natural vibrations and seismic analysis of large structures in SCAD software," in *Proc. CMM-2003 – Computer Methods in Mechanics*, Gliwice, Poland, 2003, pp. 129 —130.

[4] S. Yu. Fialko, "Iterative methods for solving large-scale problems of structural mechanics using multi-core computers," Archieves of Civil and Mechanical Engineering, vol. 14, pp. 190 – 203, 2014, http:// doi:10.1016/j.acme.2013.05.009.

[5] S. Yu. Fialko, F. Żegleń, "Block Preconditioned Conjugate Gradient Method for Extraction of Natural Vibration Frequencies in Structural Analysis", Proceedings of the FedCSIS. Łódź, 2015. IEEE Xplore Digital Library, pp. 655 – 662. DOI: 10.15439/2015F87. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7321505&tag=1.

[6] S. Yu. Fialko, "PARFES: A method for solving finite element linear equations on multi-core computers," Advances in Engineering software, vol. 40, no. 12, pp. 1256-1265, 2010, http:// doi:10.1016/j.advengsoft.2010.09.002.

[7] S. Yu. Fialko, "Parallel direct solver for solving systems of linear equations resulting from finite element method on multi-core desktops and workstations", Computers and Mathematics with Applications 70, pp. 2968–2987, 2015. doi:10.1016/j.camwa.2015.10.009

[8] C. K. Gan, P. D. Haynes and M. C. Payne, "Preconditioned conjugate gradient method for sparse generalized eigenvalue problem in electronic structure calculations," *Computer Physics Communications*, vol 134, nr. 1, pp. 33 – 40, 2001, http://DOI: 10.1016/S0010-4655(00)00188-0.

[9] V. Hernbadez, J. E. Roman, A. Tomas and V. Vidal, "A survey a software for sparse eigenvalue problems," Universitat Politecnica De Valencia, SLEPs technical report STR-6, 2009.

[10] A. V. Knyazev and K. Neymayr, "Efficient solution of symmetric eigenvalue problem using multigrid preconditioners in the locally optimal block conjugate gradient method," *Electronic Transactions on Numerical Analysis*, vol. 15, pp. 38 – 55, 2003.

[11] A. V. Knyazev, M. E. Argentati, I. Lashuk, E.E. Ovtchinnikov, "Block Locally Optimal Preconditioned Eigenvalue Xolvers (BLOPEX) in HYPRE and PETSC". URL: http://arxiv.org/pdf/0705.2626.pdf..

[12] Y. Saad, *Numerical methods for large eigenvalue problems, Revised edition, Classics in applied mathematics*. SIAM, 2011, http://dx.doi.org/10.1137/1.9781611970739.

[13] Intel Math Kernel Library Reference Manual. URL: https://software.intel.com/ru-ru/node/521001 (Last access: 20.06.2016).