On a Multigrid Eigensolver for Linear Elasticity Problems

Maxim Larin

Institut für Geometrie und Praktische Mathematik,
Rheinisch-Wesfälische Technische Hochschule Aachen,
Templergraben 55, D-52056 Aachen, Germany,
larin@igpm.rwth-aachen.de

Abstract. In the early eighties the direct application of a multigrid technique for solving the partial eigenvalue problem of computing few of the smallest eigenvalues and their corresponding eigenvectors of a differential operator was proposed by Brandt, McCormick and Ruge [1]. In the present paper an experimental study of the method for model linear elasticity problems is carried out. Based on these results we give some practical advice for a good choice of multigrid-related parameters.

1 Introduction

Eigenvalue problems for differential equations occur in many branches of science and engineering such as mechanics, chemistry, acoustics or optics. The usual methods for solving eigenvalue problems are based on the effect of domination of the smallest eigenvalues in result of repeated multiplication of the inverse matrix A^{-1} on a vector. This applies to such popular techniques as a subspace iteration, the Rayleigh quotient and the Lanczos method [5]. However, with large–scale finite element problems it is often desirable to avoid a costly inversion or, to be more precise, exact factorization of the matrix A . The simplest way is to use some preconditioned iterative procedure instead of the direct method for solving the linear system with A whenever it is required in the algorithm. In particular, multigrid methods allow us to construct *optimal* preconditioners for a sufficiently wide number of industrial applications [6].

Today among of a lot of literature published about efficient eigensolvers one can extract two main directions. The first one is based on the application of multigrid methods to construct an approximated inverse of A and use them as a preconditioner, whereas the second one directly applies the main multigrid ideas of the fine grid relaxation and the coarse grid correction (see [2, 3] and references therein).

However, all of these methods treat this eigenvalue problem purely algebraic, and hence, we lose some valuable information about the desired eigenpairs. For example, the smallest eigenvectors of the Laplace operator are very smooth, i.e. they can be well approximated on coarser grids, and hence, one can use this information during the solution process. The nested iteration multigrid process,

which takes the advantage of this smoothness, was proposed by Brandt, McCormick and Ruge [1]. The idea of this method is that the eigenvalue problems are solved on the sequence of finer grids using an interpolant of the solution on each level as the initial guess for the next one and improving it by an inner nonlinear multigrid method, i.e. suppressing the high frequency oscillations arising as a result of the interpolation process. The present work is an experimental study to understand the main points of the method, the accuracy of computed eigenpairs and regarding its computational complexity.

The paper is organized as follows. In Section 2 the formal definition of the linear elasticity eigenvalue problem is given. The full multigrid method for finding p smallest eigenvalues and their eigenvectors or, shortly, the FMG-EV(p) method will be discussed in Section 3. Based on numerical results we will make some recommendations for chosing effective user-defined parameters.

2 Problem formulation

The linear elasticity eigenvalue problem in a domain Ω in $\mathbf{R}^3$ with boundary Γ can be formulated in terms of the displacement vector $\overline{\mathbf{u}} = (u_1, u_2, u_3)$, stress tensor $\sigma = (\sigma_{ij})$ and strain tensor $\epsilon = (\epsilon_{ij})$ as follows

$$\begin{aligned} &div\,\sigma = \lambda\overline{\mathbf{u}}, \qquad \epsilon_{ij} = \frac{1}{2}\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big), \qquad \sigma_{ij}(\mathbf{u}) = \sum_{k,l=1}^{3} c_{ijkl}(x)\,\epsilon_{kl}(\overline{\mathbf{u}}), \\ &\overline{\mathbf{u}} = 0 \text{ on } \Gamma_D, \qquad \sigma\cdot\overline{n} = 0 \text{ on } \Gamma_N = \Gamma\backslash\Gamma_D. \end{aligned} \tag{1}$$

Here $c_{ijkl}(x)$ is elasticity tensor depending on positive material (Lamé) coefficients, $\overline{n}$ is the outward pointing normal on Γ_N and Γ_D, Γ_N denote Dirichlet and Neumman parts of the boundary, respectively. We assume that $\Gamma_D \neq \emptyset$ and the Korn inequality holds. These assumptions are needed to ensure that all eigenvalues λ of the problem (1) are positive. For more details regarding the problem formulation we refer to [4] or other books on elasticity.

The variational formulation of the linear elasticity eigenvalue problem (1) can be derived in a standard way and is stated as follows:

Find $\lambda_k \in \mathbf{R}$ and $\mathbf{u}_k \in [H_0^1(\Omega)]^3 = \{\mathbf{v} \in [H^1(\Omega)]^3 \,:\, \mathbf{v} = 0 \text{ on } \Gamma_D\}$ such that

$$\begin{aligned} &a(\mathbf{u}_k, \mathbf{v}) = \lambda_k b(\mathbf{u}_k, \mathbf{v}), \; k = 1, 2, \ldots, p, \ldots \\ &b(\mathbf{u}_k, \mathbf{u}_l) = \delta_{kl}, \qquad\quad l \;\; = 1, 2, \ldots, p, \ldots \end{aligned} \tag{2}$$

for all $\mathbf{v} \in [H_0^1(\Omega)]^3$, where $H^1(\Omega)$ is the usual Sobolev space, δ_{ij} is the Kronecker symbol and two bilinear forms $a(\mathbf{u}, \mathbf{v})$ and $b(\mathbf{u}, \mathbf{v})$ are defined by

$$a(\mathbf{u}, \mathbf{v}) = \int_\Omega \left[\sum_{i,j,k,l=1}^{3} c_{ijkl}\, \frac{\partial u_i}{\partial x_j} \frac{\partial v_k}{\partial x_l} \right] d\Omega, \qquad b(\mathbf{u}, \mathbf{v}) = \int_\Omega \left[\sum_{i=1}^{3} u_i v_i \right] d\Omega.$$

Let us assume that the domain Ω is decomposed by a set of finite elements Υ. Introducing $V_h \subset H^1(\Omega)$ the space of vector functions with local support,

associated with the vertices of Υ, and applying to (2) the standard Galerkin procedure, we obtain the following eigenvalue problem

$$\begin{array}{l} A\,\mathbf{u}_k = \lambda_k \mathbf{u}_k, \quad A = A^T \in \mathbf{R}^{n\times n}, \quad 0 < \lambda_1 \le \lambda_2 \le \ldots \le \lambda_p, \\ (\mathbf{u}_k, \mathbf{u}_l) = \delta_{kl}, \quad k,l = 1,\ldots,p, \end{array} \tag{3}$$

where the matrix A corresponds to the bilinear form $a(\cdot,\cdot)$ and $(\mathbf{u},\mathbf{v}) = \mathbf{u}^T\mathbf{v}$.

3 The FMG-EV(p) method

Let $\Upsilon_L \subset \Upsilon_{L-1} \subset \ldots \subset \Upsilon_1 \subset \Upsilon_0 = \Upsilon$ be a sequence of nested finite element grids, such that all grids have a similar structure and the (fine) grid Υ_k is defined from the (coarse) grid Υ_{k+1} by an uniform refinement in all spatial directions. The Galerkin procedure applied on each grid leads to the sequence of eigenvalue problems with the corresponding matrices $A^{(k)}$ of the decreasing order. Moreover, the standard (geometrical) linear restriction R_k^{k+1} and interpolation P_{k+1}^k operators are defined.

$\{\mu_i^{(0)}, \mathbf{v}_i^{(0)}\}_{i=1}^p$ = **FMG-EV**$(p, A^{(L)}, \ldots, A^{(0)})$:
  $i = 0,\ k = L,\ \mu_{i-1}^{(k)} = 0$
1 $i = i + 1$
  Compute the coarse level approximation $\{\mu_i^{(k)}, \mathbf{v}_i^{(k)}\}$,
  which is orthogonal to computed ones $\{\mu_j^{(k)}, \mathbf{v}_j^{(k)}\}_{j=1}^{i-1}$
  if $((i < cn_k)$ **and** $(i < p))$ go to 1
  $\{\mu_j^{(k)}, \mathbf{v}_j^{(k)}\}_{j=1}^{i}$ = **RITZ** $(A^{(0)}, \{\mathbf{v}_j^{(k)}\}_{j=1}^{i})$
2 **if** $(k = 0)$ **stop**
  $k = k - 1$
  for $s = 1, \ldots, i$
    $\mathbf{v}_{s,0}^{(k)} = P_{k+1}^k \mathbf{v}_s^{(k+1)}$
    $\mu_{s,0}^{(k)} = \mu_s^{(k+1)}$
    for $it = 1, \ldots, q$ (*inner iteration*)
      $\{\mu_{s,it}^{(k)}, \mathbf{v}_{s,it}^{(k)}\}$ = **FAS-EV** $(A^{(k)}, \mu_{s,it-1}^{(k)}, \mathbf{v}_{s,it-1}^{(k)}, \{\mathbf{v}_j^{(k)}\}_{j=1}^{s-1})$
  $\{\mu_j^{(k)}, \mathbf{v}_j^{(k)}\}_{j=1}^{i}$ = **RITZ** $(A^{(0)}, \{\mathbf{v}_{j,q}^{(k)}\}_{j=1}^{i})$
  if $(i < p)$ **then** go to 1
  else go to 2

Assume that we have an approximated solution $\mathbf{v}_i^{(k+1)}$ on the (coarse) grid and using their interpolants $\mathbf{v}_{i,0}^{(k)} = P_{k+1}^k \mathbf{v}_i^{(k+1)}$ as the initial guess for some inner iterative process on the next (fine) grid. Then, to eliminate the high frequency oscillations, which arise as a result of the interpolation process, we apply q times the inner multigrid method, which is based on the Full Approximation Scheme (FAS) approach and used the same sequence of matrices $\{A^{(k)}\}$. Following [1] we proceed vector-by-vector through the inner nonlinear method using

an orthogonalization condition to keep them separate. Usually, one or two inner multigrid sweeps are enough to suppress all (or nearly all) undesired frequencies in the approximated solution. It is confirmed by numerical results. Finally, we improve the eigenvector approximations by the Rayleigh-Ritz projection method and repeat this process on the next finer grid until the finest grid is reached.

$\{\mu_i^{(l)}, \mathbf{v}_i^{(l)}\}$ = **FAS-EV** $(A^{(l)}, \mu_{i,0}^{(l)}, \mathbf{v}_{i,0}^{(l)}, \{\mathbf{v}_j^{(k)}\}_{j=1}^{i-1})$:
  $\mathbf{b}_i^{(l)} = \mathbf{0}$
  for $k = l, \ldots, L-1$
    for $it = 1, \ldots, \nu_1$ $\quad$ (*presmoothing*)
      $\{\mu_{i,it}^{(k)}, \mathbf{v}_{i,it}^{(k)}\}$ = **RELAX** $(A^{(k)}, \mu_{i,it-1}^{(k)}, \mathbf{v}_{i,it-1}^{(k)}, \{\mathbf{v}_j^{(k)}\}_{j=1}^{i-1}, \mathbf{b}_i^{(k)})$
    $\{\mu_i^{(k)}, \mathbf{v}_i^{(k)}\} = \{\mu_{i,\nu_1}^{(k)}, \mathbf{v}_{i,\nu_1}^{(k)}\}$
    $\mathbf{b}_i^{(k+1)} = R_k^{k+1}\mathbf{b}_i^{(k)} + \left(A^{(k+1)}R_k^{k+1} - R_k^{k+1}A^{(k)}\right)\mathbf{v}_i^{(k)}$
    $\mathbf{v}_{i,0}^{(k+1)} = R_k^{k+1}\mathbf{v}_i^{(k)}$
    $\mu_{i,0}^{(k+1)} = \mu_i^{(k)}$
  Solve the coarse level problem $A^{(L)}\mathbf{v}_i^{(L)} = \mu_i^{(L)}\mathbf{v}_i^{(L)} + \mathbf{b}_i^{(L)}$
  with $(\mathbf{v}_i^{(L)}, R_l^L\mathbf{v}_j^{(l)}) = (R_{L-1}^L\mathbf{v}_i^{(L-1)}, R_l^L\mathbf{v}_j^{(l)}), j = 1, \ldots, i-1$
  for $k = L-1, \ldots, l$
    $\mathbf{v}_{i,0}^{(k)} = \mathbf{v}_i^{(k)} + P_{k+1}^k(\mathbf{v}_i^{(k+1)} - R_k^{k+1}\mathbf{v}_i^{(k)})$
    $\mu_{i,0}^{(k)} = \mu_i^{(k+1)}$
    for $it = 1, \ldots, \nu_2$ $\quad$ (*postsmoothing*)
      $\{\mu_{i,it}^{(k)}, \mathbf{v}_{i,it}^{(k)}\}$ = **RELAX** $(A^{(k)}, \mu_{i,it-1}^{(k)}, \mathbf{v}_{i,it-1}^{(k)}, \{\mathbf{v}_j^{(k)}\}_{j=1}^{i-1}, \mathbf{b}_i^{(k)})$
    $\{\mu_i^{(k)}, \mathbf{v}_i^{(k)}\} = \{\mu_{i,\nu_2}^{(k)}, \mathbf{v}_{i,\nu_2}^{(k)}\}$

There are two main problems with this approach. First of all, there is no exact correspondence between eigenvectors on different levels, i.e. the i-th eigenvalue and its eigenvector on the coarse level can be a *good* approximation to j-th eigenpair on the fine level with $i \neq j$. This is not a problem if j is less than p, and hence, during the following Ritz projection we find both of the desired eigenvectors on the fine level. However, if j is greater than p, then the further efforts to improve this approximation in the direction of i-th eigenvector are vain since FAS works good only in a small neighbourhood of the solution [6]. Thus, on the coarse level we can find a fixed number of approximation vectors, which are a good approximation to desired eigenvectors. This number depends on the order of the coarse level problem, and hence, can be defined as cn_L, where c is a coefficient less than 1. Now if $p > cn_L$, then we find the coarse level approximations only for first cn_L eigenvectors and start our nested iteration process with this smaller number of vectors. After transfering and processing these approximated eigenvectors on the next finer level we compute the deficient coarse level approximations using the relaxation steps followed by orthogonalization with respect to previous ones, and if p is still greater than cn_{L-1}, then we repeat this process on the next finer level and so on. Note that the coarsest level used in the inner multilevel method for ith eigenvector is one on each $\mathbf{v}_i^{(k)}$ first appeared.

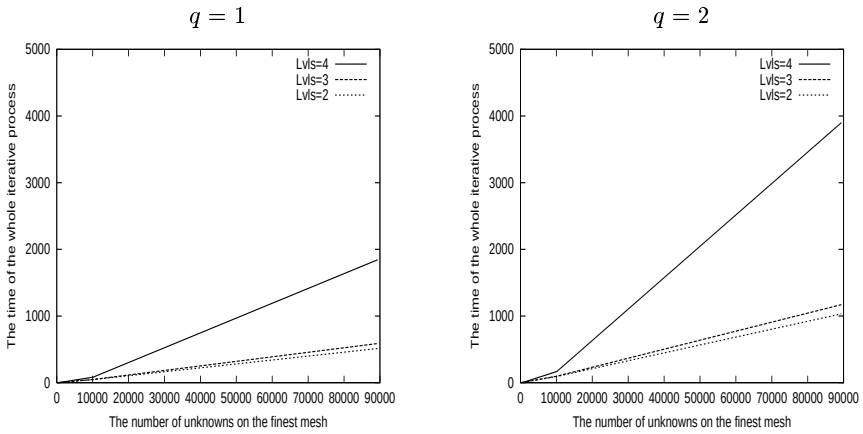

Fig. 1. The time of the whole iterative process **vs.** the number of unknowns.

Secondly, since we apply the inner nonlinear multigrid method sequentially to each approximated eigenvector, then we have to orthogonalize the current eigenvector approximation to previously computed ones. On the other side, the main FAS idea is in the construction of the coarse grid problem so that its solution is the fine level solution transfered to the coarse grid. Using it we obtain the following sequence of problems on level l

$$\begin{aligned} &A^{(k)}\mathbf{w}_i^{(k)} = \hat{\mu}_i^{(k)}\mathbf{w}_i^{(k)} + \mathbf{b}_i^{(k)},\ \mathbf{b}_i^{(k)} = R_{k-1}^k\mathbf{b}_i^{(k-1)} + \left(A^{(k)}R_{k-1}^k - R_{k-1}^k A^{(k-1)}\right)\mathbf{v}_i^{(k-1)}, \\ &\phi_k(\mathbf{w}_i^{(k)}) = \sigma_k, \qquad\qquad \mathbf{b}_i^{(l)} = 0, \qquad\qquad k = l, l+1, \ldots, L, \quad i = 1, \ldots, p. \end{aligned} \tag{4}$$

where $\phi_k(\cdot)$ is a normalization condition, which guarantees the uniqueness of (4), and specifies the size of the solutions σ_k. Note that it is not actually necessary that σ_k be equal to 1 since a solution of any reasonable size is acceptable. The specific nature of $\phi_k(\cdot)$ will be discussed below.

Let $\mathbf{u}_j^{(l)}$ is the solution on level l, then $R_l^{l+1}\mathbf{u}_j^{(l)}$ is the solution on next level, and hence, if we want to find $\mathbf{u}_i^{(l)}$, $i \neq j$, we have to orthogonalize the current eigenvector approximation $\mathbf{v}_i^{(l+1)}$ to $R_l^{l+1}\mathbf{u}_j^{(l)}$. Unfortunately, we do not know the exact solution $\mathbf{u}_j^{(l)}$ (it is our original problem!), but we have $\mathbf{v}_j^{(l)}$, $j < i$, which is a good approximation to $\mathbf{u}_j^{(l)}$. Thus, from a practical point of view $R_l^{l+1}\mathbf{v}_j^{(l)}$ is an approximation to $R_l^{l+1}\mathbf{u}_j^{(l)}$, which one can use to keep orthogonalization on the level $l+1$. Now we define

$$\phi_k(\mathbf{v}_i^{(k)}) = (\mathbf{v}_i^{(k)}, R_l^k\mathbf{v}_j^{(l)}), \quad \sigma_k = (R_{k-1}^k\mathbf{v}_i^{(k)}, R_l^k\mathbf{v}_j^{(l)}) \quad j = 1, \ldots, i. \tag{5}$$

Nevertheless, we have to note that the condition (5) does not guarantee a uniqueness of the solution (4), since we work with approximations rather than exact solutions. In our case it may happen that the sequence of test vectors $\{R_l^k\mathbf{v}_j^{(l)}\}_{j=1}^{i-1}$

used for the orthogonalization can be linear dependent, and hence, on the level k we will compute $\mathbf{v}_i^{(k)}$, which is a good approximation to $\mathbf{v}_j^{(l)}$, $j < i$, and which we already know, instead the desired approximation to $\mathbf{v}_i^{(l)}$. It is confirmed by numerical results.

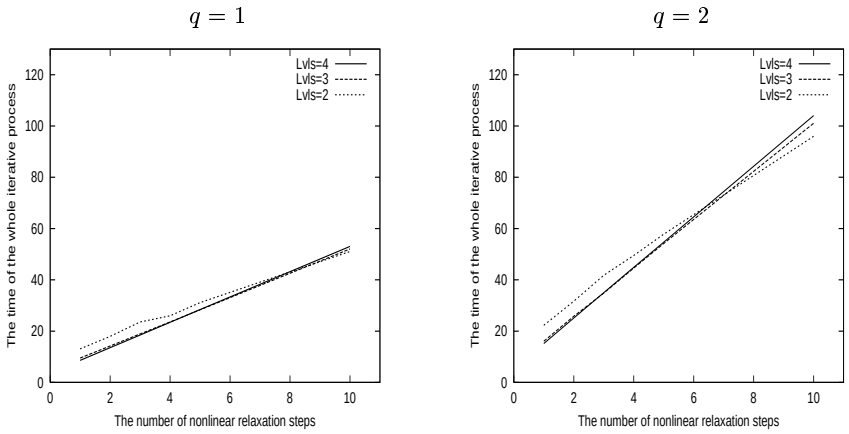

Fig. 2. The time of the whole iterative process **vs.** the number of smoothing steps.

Table 1. Accuracy of the computed eigenpair $\{\mu_1, \mathbf{v}_1\}$ as a function of ν and q, $p = 1$

N=16		$q = 1$			$q = 2$		
Lvls	ν	$\lvert\mu_1 - \lambda_1\rvert$	$\lVert\mathbf{A}\mathbf{v}_1 - \mu_1\mathbf{v}_1\rVert$	Time	$\lvert\mu_1 - \lambda_1\rvert$	$\lVert\mathbf{A}\mathbf{v}_1 - \mu_1\mathbf{v}_1\rVert$	Time
4	1	$0.18766 \cdot 10^{-2}$	$0.37333 \cdot 10^{-1}$	8.60	$0.13093 \cdot 10^{-3}$	$0.89867 \cdot 10^{-2}$	15.17
	2	$0.13861 \cdot 10^{-3}$	$0.90307 \cdot 10^{-2}$	13.56	$0.13880 \cdot 10^{-5}$	$0.67454 \cdot 10^{-3}$	25.12
	3	$0.90189 \cdot 10^{-5}$	$0.19036 \cdot 10^{-2}$	18.46	$0.18045 \cdot 10^{-7}$	$0.75458 \cdot 10^{-4}$	34.99
	4	$0.13531 \cdot 10^{-5}$	$0.67833 \cdot 10^{-3}$	23.45	$0.13159 \cdot 10^{-8}$	$0.15376 \cdot 10^{-4}$	44.89
	5	$0.11025 \cdot 10^{-6}$	$0.19556 \cdot 10^{-3}$	28.42	$0.25888 \cdot 10^{-9}$	$0.55911 \cdot 10^{-5}$	54.69
	10	$0.14618 \cdot 10^{-8}$	$0.12328 \cdot 10^{-4}$	53.03	$0.34772 \cdot 10^{-11}$	$0.65020 \cdot 10^{-6}$	104.07
3	1	$0.19283 \cdot 10^{-2}$	$0.39311 \cdot 10^{-1}$	9.50	$0.10903 \cdot 10^{-3}$	$0.81701 \cdot 10^{-2}$	16.04
	2	$0.12757 \cdot 10^{-3}$	$0.85886 \cdot 10^{-2}$	14.19	$0.18321 \cdot 10^{-5}$	$0.78770 \cdot 10^{-3}$	25.79
	3	$0.13519 \cdot 10^{-4}$	$0.23281 \cdot 10^{-2}$	18.89	$0.72569 \cdot 10^{-7}$	$0.10573 \cdot 10^{-3}$	34.82
	4	$0.18275 \cdot 10^{-5}$	$0.77967 \cdot 10^{-3}$	23.55	$0.21767 \cdot 10^{-7}$	$0.26623 \cdot 10^{-4}$	44.57
	5	$0.28713 \cdot 10^{-6}$	$0.24979 \cdot 10^{-3}$	28.32	$0.10061 \cdot 10^{-7}$	$0.16873 \cdot 10^{-4}$	54.17
	10	$0.36640 \cdot 10^{-7}$	$0.32665 \cdot 10^{-4}$	52.02	$0.54730 \cdot 10^{-9}$	$0.37209 \cdot 10^{-5}$	101.09
2	1	$0.18780 \cdot 10^{-2}$	$0.39511 \cdot 10^{-1}$	13.08	$0.11463 \cdot 10^{-3}$	$0.84105 \cdot 10^{-2}$	22.25
	2	$0.15070 \cdot 10^{-3}$	$0.97192 \cdot 10^{-2}$	17.95	$0.19856 \cdot 10^{-5}$	$0.75905 \cdot 10^{-3}$	31.70
	3	$0.19502 \cdot 10^{-4}$	$0.25037 \cdot 10^{-2}$	23.58	$0.17259 \cdot 10^{-5}$	$0.21648 \cdot 10^{-3}$	41.76
	4	$0.21043 \cdot 10^{-5}$	$0.81073 \cdot 10^{-3}$	25.97	$0.10470 \cdot 10^{-5}$	$0.15833 \cdot 10^{-3}$	49.52
	5	$0.13009 \cdot 10^{-5}$	$0.32103 \cdot 10^{-3}$	31.10	$0.11669 \cdot 10^{-5}$	$0.17195 \cdot 10^{-3}$	57.79
	10	$0.15593 \cdot 10^{-5}$	$0.19889 \cdot 10^{-3}$	51.08	$0.51628 \cdot 10^{-6}$	$0.10705 \cdot 10^{-3}$	96.00

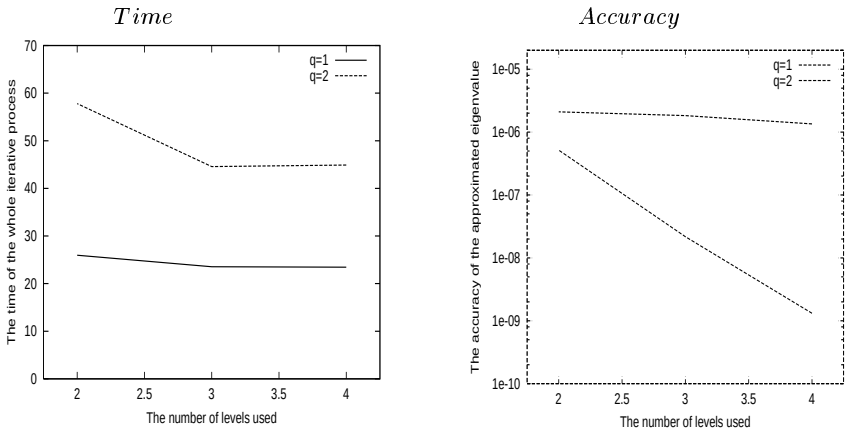

Fig. 3. The time of the iterative process and the accuracy **vs.** the number of levels.

4 Numerical results

To test the method we first consider the eigenvalue problem (3) for finding the smallest ($p = 1$) eigenpair $\{\lambda_1, \mathbf{u}_1\}$. We assume the piecewise–linear finite–element discretization of the elasticity problem (1) in the cubical domain $\Omega = [0,1]^3$ on a uniform Cartesian mesh $\mathcal{T}_h = N \times N \times N$ with meshsize $h = N^{-1}$. Consider homogeneous Dirichlet boundary conditions, i.e., $\Gamma_D = \Gamma, \Gamma_N = \emptyset$.

In all tables of this paper $\nu = \nu_1 = \nu_2$ is the number of pre- and post-smoothing iterations on each level, q is the number of inner FAS-EV iterations applied on each level, $Lvls$ is the number of levels used and $Time$ is the CPU time of the whole iterative process. All calculations were performed on an IBM-SP2 in double precision. By μ_i and $\mathbf{v}_i$ we denote the approximation to the exact eigenvalues and eigenvectors, λ_i and $\mathbf{u}_i$, respectively. Correspondingly, $|\mu_i - \lambda_i|$ measures the difference between the exact and the computed eigenvalues and $\|\mathbf{A}\mathbf{v}_i - \mu_i \mathbf{v}_i\|$ measures the quality of the approximated pair $\{\mu_i, \mathbf{v}_i\}$. The exact eigenvalues λ_i have been computed by the Gaus-Seidel iteration method followed by standard Gramm-Schmidt orthogonalization process until the following stopping criterion was satisfied

$$\|\mathbf{r}_i^{(k)}\| / \|\mathbf{r}_i^{(0)}\| < 10^{-12}, \quad \mathbf{r}_i^{(k)} = A\mathbf{v}_i^{(k)} - \mu_i \mathbf{v}_i^{(k)},$$

where $\mathbf{r}_i^{(0)}$ and $\mathbf{r}_i^{(k)}$ are the initial and the final residuals, respectively.

From the numerical results in Table 1 one can see that by increasing the number of relaxation steps ν per level without changing other parameters, the accuracy of the approximated solution is improved. Comparing results for $q = 1$ with corresponding ones for $q = 2$ in Table 1 one can also see that increasing the number of inner FAS-EV steps leads to an improved accuracy of the approximated eigenpair $\{\mu_i, \mathbf{v}_i\}$. On the other hand, the accuracy of the approximated

solution does not depend sensitively on the number of levels used. Due to the discretization error for this problem is $O(h^2) = O(10^{-3})$ one can see that the FMG-EV(p) method can compute the first eigenvalue with a reasonable computational cost, when $q = 1$ and $\nu = 3$.

Table 2. Accuracy of computed eigenvalues μ_i as a function of ν and q, $p = 5$

q	ν	$\lvert\mu_1 - \lambda_1\rvert$	$\lvert\mu_2 - \lambda_2\rvert$	$\lvert\mu_3 - \lambda_3\rvert$	$\lvert\mu_4 - \lambda_4\rvert$	$\lvert\mu_5 - \lambda_5\rvert$
Lvls = 4, NFine = 10125, NCoarse = 3						
1	1	$0.17121 \cdot 10^{-2}$	$0.18504 \cdot 10^{-2}$	$0.23162 \cdot 10^{-2}$	$0.21263 \cdot 10^{-2}$	$0.24166 \cdot 10^{-2}$
	2	$0.87427 \cdot 10^{-4}$	$0.18120 \cdot 10^{-3}$	$0.18459 \cdot 10^{-3}$	$0.14426 \cdot 10^{-3}$	$0.16316 \cdot 10^{-3}$
	3	$0.59710 \cdot 10^{-5}$	$0.15262 \cdot 10^{-4}$	$0.23290 \cdot 10^{-4}$	$0.16597 \cdot 10^{-4}$	$0.20785 \cdot 10^{-4}$
	4	$0.64233 \cdot 10^{-6}$	$0.18220 \cdot 10^{-5}$	$0.27387 \cdot 10^{-5}$	$0.22954 \cdot 10^{-5}$	$0.26639 \cdot 10^{-5}$
	5	$0.62436 \cdot 10^{-7}$	$0.27586 \cdot 10^{-6}$	$0.33646 \cdot 10^{-6}$	$0.29719 \cdot 10^{-6}$	$0.81938 \cdot 10^{-6}$
	10	$0.12715 \cdot 10^{-8}$	$0.16685 \cdot 10^{-8}$	$0.54289 \cdot 10^{-8}$	$0.13347 \cdot 10^{-6}$	$0.34387 \cdot 10^{-6}$
2	1	$0.66509 \cdot 10^{-4}$	$0.13746 \cdot 10^{-3}$	$0.14220 \cdot 10^{-3}$	$0.11944 \cdot 10^{-3}$	$0.13899 \cdot 10^{-3}$
	2	$0.56575 \cdot 10^{-6}$	$0.17661 \cdot 10^{-5}$	$0.25094 \cdot 10^{-5}$	$0.16908 \cdot 10^{-5}$	$0.20703 \cdot 10^{-5}$
	3	$0.52939 \cdot 10^{-8}$	$0.36231 \cdot 10^{-7}$	$0.47925 \cdot 10^{-7}$	$0.13661 \cdot 10^{-6}$	$0.40596 \cdot 10^{-6}$
	4	$0.52667 \cdot 10^{-9}$	$0.87679 \cdot 10^{-9}$	$0.23866 \cdot 10^{-8}$	$0.69039 \cdot 10^{-7}$	$0.18090 \cdot 10^{-6}$
	5	$0.92139 \cdot 10^{-10}$	$0.16693 \cdot 10^{-9}$	$0.27081 \cdot 10^{-8}$	$0.17478 \cdot 10^{-7}$	$0.11907 \cdot 10^{-6}$
	10	$0.11360 \cdot 10^{-9}$	$0.11232 \cdot 10^{-9}$	$0.42050 \cdot 10^{-10}$	$0.65845 \cdot 10^{-8}$	$0.18503 \cdot 10^{-7}$
Lvls = 3, NFine = 10125, NCoarse = 81						
1	1	$0.17126 \cdot 10^{-2}$	$0.18520 \cdot 10^{-2}$	$0.23164 \cdot 10^{-2}$	$0.21153 \cdot 10^{-2}$	$0.22651 \cdot 10^{-2}$
	2	$0.86727 \cdot 10^{-4}$	$0.18121 \cdot 10^{-3}$	$0.18583 \cdot 10^{-3}$	$0.15580 \cdot 10^{-3}$	$0.16274 \cdot 10^{-3}$
	3	$0.59921 \cdot 10^{-5}$	$0.15343 \cdot 10^{-4}$	$0.23375 \cdot 10^{-4}$	$0.12978 \cdot 10^{-4}$	$0.16919 \cdot 10^{-4}$
	4	$0.63646 \cdot 10^{-6}$	$0.18893 \cdot 10^{-5}$	$0.27678 \cdot 10^{-5}$	$0.22854 \cdot 10^{-5}$	$0.28454 \cdot 10^{-5}$
	5	$0.70249 \cdot 10^{-7}$	$0.27376 \cdot 10^{-6}$	$0.34354 \cdot 10^{-6}$	$0.38226 \cdot 10^{-6}$	$0.91077 \cdot 10^{-6}$
	10	$0.13113 \cdot 10^{-8}$	$0.20920 \cdot 10^{-8}$	$0.71929 \cdot 10^{-8}$	$0.16827 \cdot 10^{-6}$	$0.23884 \cdot 10^{-6}$
2	1	$0.66521 \cdot 10^{-4}$	$0.13756 \cdot 10^{-3}$	$0.14223 \cdot 10^{-3}$	$0.11527 \cdot 10^{-3}$	$0.13875 \cdot 10^{-3}$
	2	$0.56377 \cdot 10^{-6}$	$0.17248 \cdot 10^{-5}$	$0.25207 \cdot 10^{-5}$	$0.15277 \cdot 10^{-5}$	$0.22036 \cdot 10^{-5}$
	3	$0.68877 \cdot 10^{-8}$	$0.36520 \cdot 10^{-7}$	$0.81525 \cdot 10^{-7}$	$0.82430 \cdot 10^{-7}$	$0.36574 \cdot 10^{-6}$
	4	$0.78704 \cdot 10^{-9}$	$0.22278 \cdot 10^{-8}$	$0.15379 \cdot 10^{-7}$	$0.86195 \cdot 10^{-7}$	$0.16180 \cdot 10^{-6}$
	5	$0.83009 \cdot 10^{-10}$	$0.23116 \cdot 10^{-9}$	$0.94830 \cdot 10^{-8}$	$0.62724 \cdot 10^{-7}$	$0.87238 \cdot 10^{-7}$
	10	$0.11371 \cdot 10^{-9}$	$0.11276 \cdot 10^{-9}$	$0.36793 \cdot 10^{-9}$	$0.69298 \cdot 10^{-8}$	$0.12069 \cdot 10^{-7}$
Lvls = 2, NFine = 10125, NCoarse = 1029						
1	1	$0.16598 \cdot 10^{-2}$	$0.17743 \cdot 10^{-2}$	$0.22529 \cdot 10^{-2}$	$0.20921 \cdot 10^{-2}$	$0.22736 \cdot 10^{-2}$
	2	$0.89283 \cdot 10^{-4}$	$0.17879 \cdot 10^{-3}$	$0.18547 \cdot 10^{-3}$	$0.14831 \cdot 10^{-3}$	$0.16975 \cdot 10^{-3}$
	3	$0.68256 \cdot 10^{-5}$	$0.15841 \cdot 10^{-4}$	$0.23290 \cdot 10^{-4}$	$0.15277 \cdot 10^{-4}$	$0.19748 \cdot 10^{-4}$
	4	$0.10038 \cdot 10^{-5}$	$0.21974 \cdot 10^{-5}$	$0.32588 \cdot 10^{-5}$	$0.17504 \cdot 10^{-5}$	$0.24626 \cdot 10^{-5}$
	5	$0.16654 \cdot 10^{-6}$	$0.33157 \cdot 10^{-6}$	$0.22759 \cdot 10^{-5}$	$0.19340 \cdot 10^{-6}$	$0.65046 \cdot 10^{-6}$
	10	$0.13450 \cdot 10^{-8}$	$0.56302 \cdot 10^{-8}$	$0.20701 \cdot 10^{-5}$	$0.18521 \cdot 10^{-7}$	$0.10047 \cdot 10^{-6}$
2	1	$0.66248 \cdot 10^{-4}$	$0.13791 \cdot 10^{-3}$	$0.14184 \cdot 10^{-3}$	$0.11067 \cdot 10^{-3}$	$0.15442 \cdot 10^{-3}$
	2	$0.56097 \cdot 10^{-6}$	$0.18273 \cdot 10^{-5}$	$0.56997 \cdot 10^{-5}$	$0.16318 \cdot 10^{-5}$	$0.25670 \cdot 10^{-5}$
	3	$0.25852 \cdot 10^{-7}$	$0.37555 \cdot 10^{-7}$	$0.17063 \cdot 10^{-5}$	$0.26033 \cdot 10^{-7}$	$0.30782 \cdot 10^{-7}$
	4	$0.13821 \cdot 10^{-8}$	$0.25116 \cdot 10^{-8}$	$0.94665 \cdot 10^{-6}$	$0.46884 \cdot 10^{-8}$	$0.39074 \cdot 10^{-7}$
	5	$0.14959 \cdot 10^{-9}$	$0.46594 \cdot 10^{-8}$	$0.15020 \cdot 10^{-5}$	$0.28829 \cdot 10^{-7}$	$0.83986 \cdot 10^{-7}$
	10	$0.11348 \cdot 10^{-9}$	$0.28841 \cdot 10^{-10}$	$0.40120 \cdot 10^{-6}$	$0.18245 \cdot 10^{-8}$	$0.48712 \cdot 10^{-8}$

Total computational times for various values of *Lvls* versus the number of unknowns on the finest mesh and the number of relaxation steps are given in

Figures 1 and 2, respectively. Based on the presented numerical results we can see that the FMG-EV(p) method has a nearly optimal computational complexity, i.e. the time of the whole iterative process does not depend on the number of unknowns on the finest level and only slightly depend on the number of levels used.

Table 3. Accuracy of computed eigenpairs $\{\mu_i, \mathbf{v}_i\}$ as a function of ν and q, $p = 5$

q	ν	$\|\mathbf{A}\mathbf{v}_1 - \mu_1\mathbf{v}_1\|$	$\|\mathbf{A}\mathbf{v}_2 - \mu_2\mathbf{v}_2\|$	$\|\mathbf{A}\mathbf{v}_3 - \mu_3\mathbf{v}_3\|$	$\|\mathbf{A}\mathbf{v}_4 - \mu_4\mathbf{v}_4\|$	$\|\mathbf{A}\mathbf{v}_5 - \mu_5\mathbf{v}_5\|$
Lvls = 4, NFine = 10125, NCoarse = 3						
1	1	$0.35633 \cdot 10^{-1}$	$0.35849 \cdot 10^{-1}$	$0.48551 \cdot 10^{-1}$	$0.40670 \cdot 10^{-1}$	$0.46340 \cdot 10^{-1}$
	2	$0.69620 \cdot 10^{-2}$	$0.11551 \cdot 10^{-1}$	$0.10216 \cdot 10^{-1}$	$0.96870 \cdot 10^{-2}$	$0.96702 \cdot 10^{-2}$
	3	$0.16195 \cdot 10^{-2}$	$0.23696 \cdot 10^{-2}$	$0.31153 \cdot 10^{-2}$	$0.25545 \cdot 10^{-2}$	$0.28774 \cdot 10^{-2}$
	4	$0.54673 \cdot 10^{-3}$	$0.73684 \cdot 10^{-3}$	$0.10033 \cdot 10^{-2}$	$0.81480 \cdot 10^{-3}$	$0.88640 \cdot 10^{-3}$
	5	$0.15609 \cdot 10^{-3}$	$0.28865 \cdot 10^{-3}$	$0.33740 \cdot 10^{-3}$	$0.27417 \cdot 10^{-3}$	$0.36506 \cdot 10^{-3}$
	10	$0.12719 \cdot 10^{-4}$	$0.14326 \cdot 10^{-4}$	$0.17823 \cdot 10^{-4}$	$0.84203 \cdot 10^{-4}$	$0.13971 \cdot 10^{-3}$
2	1	$0.61016 \cdot 10^{-2}$	$0.89436 \cdot 10^{-2}$	$0.98460 \cdot 10^{-2}$	$0.83752 \cdot 10^{-2}$	$0.91747 \cdot 10^{-2}$
	2	$0.51795 \cdot 10^{-3}$	$0.72641 \cdot 10^{-3}$	$0.96730 \cdot 10^{-3}$	$0.71808 \cdot 10^{-3}$	$0.83226 \cdot 10^{-3}$
	3	$0.42454 \cdot 10^{-4}$	$0.10305 \cdot 10^{-3}$	$0.11552 \cdot 10^{-3}$	$0.11879 \cdot 10^{-3}$	$0.17028 \cdot 10^{-3}$
	4	$0.86842 \cdot 10^{-5}$	$0.15120 \cdot 10^{-4}$	$0.19860 \cdot 10^{-4}$	$0.59069 \cdot 10^{-4}$	$0.10028 \cdot 10^{-3}$
	5	$0.53047 \cdot 10^{-5}$	$0.55883 \cdot 10^{-5}$	$0.10770 \cdot 10^{-4}$	$0.31108 \cdot 10^{-4}$	$0.77286 \cdot 10^{-4}$
	10	$0.87844 \cdot 10^{-6}$	$0.14166 \cdot 10^{-5}$	$0.19605 \cdot 10^{-5}$	$0.16655 \cdot 10^{-4}$	$0.31198 \cdot 10^{-4}$
Lvls = 3, NFine = 10125, NCoarse = 81						
1	1	$0.35638 \cdot 10^{-1}$	$0.35845 \cdot 10^{-1}$	$0.48545 \cdot 10^{-1}$	$0.39561 \cdot 10^{-1}$	$0.45533 \cdot 10^{-1}$
	2	$0.69323 \cdot 10^{-2}$	$0.11575 \cdot 10^{-1}$	$0.10219 \cdot 10^{-1}$	$0.97138 \cdot 10^{-2}$	$0.98802 \cdot 10^{-2}$
	3	$0.16205 \cdot 10^{-2}$	$0.23698 \cdot 10^{-2}$	$0.31158 \cdot 10^{-2}$	$0.22178 \cdot 10^{-2}$	$0.26073 \cdot 10^{-2}$
	4	$0.54086 \cdot 10^{-3}$	$0.74379 \cdot 10^{-3}$	$0.10048 \cdot 10^{-2}$	$0.88062 \cdot 10^{-3}$	$0.81750 \cdot 10^{-3}$
	5	$0.15830 \cdot 10^{-3}$	$0.28768 \cdot 10^{-3}$	$0.33659 \cdot 10^{-3}$	$0.28437 \cdot 10^{-3}$	$0.34140 \cdot 10^{-3}$
	10	$0.12587 \cdot 10^{-4}$	$0.16523 \cdot 10^{-4}$	$0.17521 \cdot 10^{-4}$	$0.98736 \cdot 10^{-4}$	$0.10736 \cdot 10^{-3}$
2	1	$0.61022 \cdot 10^{-2}$	$0.89445 \cdot 10^{-2}$	$0.98459 \cdot 10^{-2}$	$0.82619 \cdot 10^{-2}$	$0.91492 \cdot 10^{-2}$
	2	$0.52500 \cdot 10^{-3}$	$0.71841 \cdot 10^{-3}$	$0.96738 \cdot 10^{-3}$	$0.66363 \cdot 10^{-3}$	$0.82692 \cdot 10^{-3}$
	3	$0.46287 \cdot 10^{-4}$	$0.10465 \cdot 10^{-3}$	$0.11551 \cdot 10^{-3}$	$0.99123 \cdot 10^{-4}$	$0.16709 \cdot 10^{-3}$
	4	$0.13839 \cdot 10^{-4}$	$0.19808 \cdot 10^{-4}$	$0.21173 \cdot 10^{-4}$	$0.66386 \cdot 10^{-4}$	$0.94793 \cdot 10^{-4}$
	5	$0.52278 \cdot 10^{-5}$	$0.62587 \cdot 10^{-5}$	$0.16354 \cdot 10^{-4}$	$0.57611 \cdot 10^{-4}$	$0.66844 \cdot 10^{-4}$
	10	$0.70692 \cdot 10^{-6}$	$0.79362 \cdot 10^{-6}$	$0.41566 \cdot 10^{-5}$	$0.19018 \cdot 10^{-4}$	$0.23745 \cdot 10^{-4}$
Lvls = 2, NFine = 10125, NCoarse = 1029						
1	1	$0.35119 \cdot 10^{-1}$	$0.35115 \cdot 10^{-1}$	$0.47902 \cdot 10^{-1}$	$0.39215 \cdot 10^{-1}$	$0.45660 \cdot 10^{-1}$
	2	$0.70299 \cdot 10^{-2}$	$0.11518 \cdot 10^{-1}$	$0.10113 \cdot 10^{-1}$	$0.96555 \cdot 10^{-2}$	$0.10124 \cdot 10^{-1}$
	3	$0.16783 \cdot 10^{-2}$	$0.23134 \cdot 10^{-2}$	$0.31116 \cdot 10^{-2}$	$0.25111 \cdot 10^{-2}$	$0.28154 \cdot 10^{-2}$
	4	$0.58759 \cdot 10^{-3}$	$0.79815 \cdot 10^{-3}$	$0.94454 \cdot 10^{-3}$	$0.69161 \cdot 10^{-3}$	$0.91486 \cdot 10^{-3}$
	5	$0.21834 \cdot 10^{-3}$	$0.33317 \cdot 10^{-3}$	$0.33348 \cdot 10^{-3}$	$0.26339 \cdot 10^{-3}$	$0.35363 \cdot 10^{-3}$
	10	$0.13607 \cdot 10^{-4}$	$0.25534 \cdot 10^{-4}$	$0.22342 \cdot 10^{-3}$	$0.30763 \cdot 10^{-4}$	$0.72578 \cdot 10^{-4}$
2	1	$0.60697 \cdot 10^{-2}$	$0.89510 \cdot 10^{-2}$	$0.98347 \cdot 10^{-2}$	$0.81740 \cdot 10^{-2}$	$0.96936 \cdot 10^{-2}$
	2	$0.52357 \cdot 10^{-3}$	$0.74812 \cdot 10^{-3}$	$0.98926 \cdot 10^{-3}$	$0.79192 \cdot 10^{-3}$	$0.90675 \cdot 10^{-3}$
	3	$0.88952 \cdot 10^{-4}$	$0.10400 \cdot 10^{-3}$	$0.21830 \cdot 10^{-3}$	$0.89080 \cdot 10^{-4}$	$0.86215 \cdot 10^{-4}$
	4	$0.15535 \cdot 10^{-4}$	$0.18265 \cdot 10^{-4}$	$0.14572 \cdot 10^{-3}$	$0.19477 \cdot 10^{-4}$	$0.45988 \cdot 10^{-4}$
	5	$0.58039 \cdot 10^{-5}$	$0.21082 \cdot 10^{-4}$	$0.18240 \cdot 10^{-3}$	$0.40298 \cdot 10^{-4}$	$0.63035 \cdot 10^{-4}$
	10	$0.59914 \cdot 10^{-6}$	$0.40553 \cdot 10^{-5}$	$0.96543 \cdot 10^{-4}$	$0.74399 \cdot 10^{-5}$	$0.14618 \cdot 10^{-4}$

Moreover, in Figure 3 we present the dependency of the time of eigenvalue solver and the accuracy of the computed eigenvalue as a function of the number of levels. Here we want to note a jumping of the accuracy for the case $q = 2$. It is shown that not all high frequencies can be eliminated during the first step of the FAS-EV method, and hence, we need the second one.

The results of the first five ($p = 5$) eigenvalues and their corresponding eigenvectors on a $(16 \times 16 \times 16)$-grid are given in Tables 2 and 3. Based on the results we conclude that the FMG-EV(p) method behaves similarly as described above for the case $p = 1$. We make only a few remarks.

First, a comparison of the accuracies in Tables 2 and 3 for the first eigenpair $\{\mu_1, \mathbf{v}_1\}$ with these shown in Table 1 shows that when more eigenvectors, whether or not they converge, are included in the process, all approximations are improved. Taking into account the whole computational complexity, which is growing rapidly when ν and q are increasing, a good choice of the parameters seems to be $q = 1$ and $\nu = 3$. This is similar to the results for $p = 1$.

Moreover, the present experiments show that the FAML-EV(p) method is capable of dealing with multiple eigenvalues. Indeed, we have $\lambda_1 = \lambda_2 = \lambda_3$ and one can see from Tables 2 and 3 that we find this eigenvalue and its eigensubspace without problems.

Finally, we want to mention a problem with the eigenvector orthogonalization for the inner nonlinear method. Indeed, when we try to solve the coarse level problem for the second, third or other eigenvectors during the FAS method with some given accuracy, we always perform the maximal number of iterations, i.e. we could not reach the desired accuracy. It seems that it is not a strong restriction on the suggested method since the method is still convergent. However, the proposed technique with corresponding block-type modifications for the inner nonlinear method can be done. The further investigation will be directed in this way.

Summarizing, we want to note that the present method can be used instead of the direct eigensolver when a small number of eigenpairs is required, i.e., $p < \sqrt{n}$. However, if we wanted to find a series of eigenvectors with $p > \sqrt{n}$, then the Lanzos method with LU-factorization would be less time consuming.

References

1. A. Brandt, S. McCormick and J. Ruge, *Multigrid methods for differential eigenproblems*, SIAM J. Sci. Stat. Comput., 4(2) (1983), pp. 244–260.
2. W. Hackbusch, *Multigrid Methods and Applications*, Springer-Verlag, Berlin - Heidelberg - New York, 1985.
3. A. V. Knyazev, *Preconditioned eigensolvers – an oxymoron?*, Electronic Transaction on Numerical Analysis, Volume 7, 1998, pp. 104–123.
4. A. I. Lur'e, *Theory of elasticity*, Moscow, Nauka, 1970.
5. B. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Inc., 1980.
6. U. Trottenberg, C. Oosterlee and A. Schüller, *Multigrid*, Academic Press, 2001.