

ANALYSIS OF FIXING NODES USED IN GENERALIZED INVERSE COMPUTATION

Pavla HRUSKOVA

Department of Applied Mathematics, Faculty of Electrical Engineering and Computer Science,
VSB–Technical University Ostrava, 17. listopadu 15, 708 33 Ostrava, Czech Republic

pavla.hruskova@vsb.cz

Abstract. *In various fields of numerical mathematics, there arises the need to compute a generalized inverse of a symmetric positive semidefinite matrix, for example in the solution of contact problems. Systems with semidefinite matrices can be solved by standard direct methods for the solution of systems with positive definite matrices adapted to the solution of systems with only positive semidefinite matrix. One of the possibilities is a modification of Cholesky decomposition using so called fixing nodes, which is presented in this paper with particular emphasis on proper definition of fixing nodes. The generalised inverse algorithm consisting in Cholesky decomposition with usage of fixing nodes is adopted from paper [1]. In [1], authors choose the fixing nodes using Perron vector of an adjacency matrix of the graph which is only a sub-optimal choice. Their choice is discussed in this paper together with other possible candidates on fixing node. Several numerical experiments including all candidates have been done. Based on these results, it turns out that using eigenvectors of Laplacian matrix provides better choice of fixing node than using Perron vector.*

Keywords

Cholesky decomposition, fixing nodes, generalised inverse, Laplacian matrix, spectral graph theory, stiffness matrix.

1. Introduction

The motivation of this paper is a computation of a generalized inverse of a symmetric positive semidefinite (SPS) matrix which arises for example in solution of contact problems. One of the possibilities is a modification of Cholesky decomposition.

For the purposes of this paper, we will consider the meshes arising during discretization of numerical prob-

lems from graph theory point of view and we will analyse these meshes as graphs.

The particular purpose of this analysis is to find certain nodes (further defined as “fixing nodes”) in meshes to reduce numerical instability in computation of a generalized inverse of semidefinite stiffness matrix.

Algorithm presented in this paper is taken from [1]. In [1], authors use the Perron vector of an adjacency matrix for the computation of fixing nodes, which is only a sub-optimal choice.

In this paper, the proper definition of fixing nodes is presented. Several candidates to fixing nodes are provided based on (spectral) graph techniques together with experimental results. Based on these results, it turns out that using eigenvectors of Laplacian matrix provides better choice of fixing node than using Perron vector (eigenvector of the adjacency matrix). The best choice of fixing node is analysed at the end of this paper.

2. Generalized Inverse Algorithm

Let us consider a system of consistent linear equations:

$$\mathbf{A}x = b, \quad (1)$$

with an SPS matrix \mathbf{A} . In case of FETI method, matrix \mathbf{A} is called the stiffness matrix.

If $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $b \in \text{Im } \mathbf{A}$, where $\text{Im } \mathbf{A}$ denotes the range of \mathbf{A} , then a solution $x = \mathbf{A}^+b + \mathbf{R}c$ of the system of linear equations Eq. (1) can be expressed by means of a (left) generalized inverse matrix, $\mathbf{A}^+ \in \mathbb{R}^{n \times n}$, \mathbf{R} is the matrix of rigid body modes, and \vec{c} is a vector of coefficients.

There are several known algorithms how to compute generalized inverse matrix. There can be used either standard direct methods or some of iterative methods

such as Cholesky decomposition, singular value decomposition, and their combinations.

In paper *Cholesky decomposition with fixing nodes to stable computation of a generalized inverse of the stiffness matrix of a floating structure* [1], authors Dostal, Kozubek, Kovar, Markopoulos, and Brzobohaty present their algorithm of computation of generalized inverse matrix based on Cholesky decomposition, adapted to the solution of systems with SPS matrix.

Generalized inverse algorithm [1] consists in a decomposition of the SPS matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$:

$$\begin{aligned} \mathbf{PAP}^T &= \begin{bmatrix} \tilde{A}_{JJ} & \tilde{A}_{JI} \\ \tilde{A}_{IJ} & \tilde{A}_{II} \end{bmatrix} = \\ &= \begin{bmatrix} L_{JJ} & O \\ L_{IJ} & I \end{bmatrix} \begin{bmatrix} L_{JJ}^T & L_{IJ}^T \\ O & S \end{bmatrix}, \end{aligned} \quad (2)$$

where $\tilde{A}_{JJ} \in \mathbb{R}^{r \times r}$ is well-conditioned regular part of $\mathbf{A} \in \mathbb{R}^{n \times n}$, $L_{JJ} \in \mathbb{R}^{r \times r}$ is a lower factor of the Cholesky factorization of \tilde{A}_{JJ} , $L_{IJ} \in \mathbb{R}^{s \times r}$, $L_{IJ} = \tilde{A}_{IJ}L_{JJ}^{-T}$, $\mathbf{S} \in \mathbb{R}^{s \times s}$ is a singular matrix, $s = n - r$ is the number of displacements corresponding to the fixing nodes, and \mathbf{P} is a permutation matrix.

Then the generalised inverse \mathbf{A}^+ is computed as:

$$\begin{aligned} \mathbf{A}^+ &= \mathbf{P}^T \begin{bmatrix} L_{JJ}^{-T} & -L_{JJ}^{-T}L_{IJ}^T S^\dagger \\ O & S^\dagger \end{bmatrix} \cdot \\ &\cdot \begin{bmatrix} L_{JJ}^{-1} & O \\ -L_{IJ}L_{JJ}^{-1} & I \end{bmatrix} \mathbf{P}, \end{aligned} \quad (3)$$

where S^\dagger denotes the Moore–Penrose generalized inverse. Permutation matrix \mathbf{P} , $\mathbf{P} = \mathbf{P}_2\mathbf{P}_1$, is computed in two steps. First, matrix \mathbf{P}_1 is found in the form:

$$\mathbf{P}_1\mathbf{A}\mathbf{P}_1^T = \begin{bmatrix} \tilde{A}_{JJ} & \tilde{A}_{JI} \\ \tilde{A}_{IJ} & \tilde{A}_{II} \end{bmatrix}, \quad (4)$$

where \tilde{A}_{JJ} is nonsingular and \tilde{A}_{II} corresponds to the degrees of freedom of the M fixing nodes.

Second, reordering algorithm on $\mathbf{P}_1\mathbf{A}\mathbf{P}_1^T$ is applied to get a permutation matrix \mathbf{P}_2 which leaves the part \tilde{A}_{II} without changes and enables the sparse Cholesky factorization of \tilde{A}_{JJ} .

3. Fixing Nodes and Center-like Points

3.1. Fixing Nodes

Finding fixing nodes effectively and accurately is the main ingredient of the generalized inverse algorithm.

The active choice is made in sense of permutation matrix \mathbf{P}_1 in Eq. (4) such that the rows/columns corresponding to those vertices marked as fixing nodes

are permuted at the end of the original stiffness matrix (bottom-right block). Also, the rigid body motions corresponding to degrees of freedom of selected fixing nodes are removed. Let us rewrite the definition of fixing node (first written in [2]).

Definition 1. *Fixing node:* Let $\mathbf{A}x = b$ be a system of linear equations arising from a finite element or finite difference discretization of the problem, such that \mathbf{A} has one-dimensional kernel (i.e. the singular part \tilde{A}_{II} in Eq. (4) is formed by one zero element).

The one-fixing node is the node that makes the regular part \tilde{A}_{JJ} of the stiffness matrix \mathbf{A} produced by the permutation \mathbf{P} nonsingular and well conditioned, i.e. permutation of this node to the last row/column of the matrix \mathbf{A} makes the condition number of the regular part \tilde{A}_{JJ} finite and sufficiently small.

The best choice of one-fixing node is the node k for which the regular part \tilde{A}_{JJ} of the stiffness matrix \mathbf{A} produced by the permutation \mathbf{P} has the minimal condition number over all \tilde{A}_{kk} (Symbol \tilde{A}_{kk} denotes the principal submatrix arising by removing k -th row and column from original matrix $\tilde{\mathbf{A}}$):

$$\text{cond}(\tilde{A}_{JJ}) = \min_{k=\{1, \dots, n\}} \text{cond}(\tilde{A}_{kk}). \quad (5)$$

Condition number $\text{cond}(\tilde{A}_{kk})$ is computed as:

$$\text{cond}(\tilde{A}_{kk}) = \frac{\lambda_{\max}(\tilde{A}_{kk})}{\lambda_{\min}(\tilde{A}_{kk})}, \quad (6)$$

where $\lambda_{\max}(\tilde{A}_{kk})$ and $\lambda_{\min}(\tilde{A}_{kk})$ denote the largest and the smallest eigenvalue of the non-singular matrix \tilde{A}_{kk} , respectively. Definition of more fixing nodes is composed accordingly.

Definition 2. *i-fixing nodes:* The i -fixing nodes (if the number i of i -fixing nodes is not important, we use only the term “fixing nodes” instead of the term “ i -fixing nodes”) are the set of i nodes that make the regular part \tilde{A}_{JJ} of the stiffness matrix A produced by the permutation \mathbf{P} nonsingular and well conditioned, i.e., permutation of these nodes to the bottommost rows, rightmost columns respectively, of the matrix \mathbf{A} makes the condition number of regular part \tilde{A}_{JJ} finite and sufficiently small (see Eq. (4)).

The best choice of i -fixing nodes is the set of i -fixing nodes for which the regular part \tilde{A}_{JJ} of the stiffness matrix \mathbf{A} produced by the permutation \mathbf{P} has the minimal condition number over all such permutations.

Further, we show techniques how to find one fixing node. One fixing node can stabilize well the solution of problems with one-dimensional kernel. Problem of finding more fixing nodes can be reduced to the problem of finding one fixing node by decomposition of

the mesh corresponding to given problem by a suitable software into several parts. The fixing nodes are then found one in each part.

In this case, it is guaranteed that we can obtain the best choice of one-fixing node in each part in sense of Definition 1 but the resulting set of i -fixing nodes of the whole problem do not need to be the best choice of i -fixing nodes in sense of Definition 2.

Definition 1 operates on condition numbers of residual matrices, thus it is not suitable to evaluate the fixing node in a reasonable time. Therefore, a fast heuristic has been looked for to provide a good approximation of the fixing node.

As the fixing nodes should lie near the “center” of the mesh [3], there are several possibilities how to define center-like points of mesh, or corresponding graph respectively.

3.2. Center-like Points

One of the approaches to find an approximation of the best choice of one-fixing node is based on the mechanical interpretation of the problem. It follows the idea of choice of the fixing node near the “center” of mesh. This translates directly into choosing one of the vertices in the center of the graph. Finding graph centers is not suitable for the numerical solution of large problems but it provides a good referential point to the other methods. This approach has been published in [3].

Another idea how to identify the fixing node is based on spectral approach, namely on eigenvector of the adjacency matrix of the graph.

Definition 3. *Capital vertex:* Let the capital vertex of a graph G be a vertex v_i that corresponds to the index i of the highest value in the Perron vector of the corresponding adjacency matrix of graph G .

The capital vertex does not identify the center of the graph, rather a vertex from which most walks of a given length can be realized. By choosing the capital vertex to be the fixing node we expect to achieve a stable numerical solution. Moreover, identifying the capital vertex is fast.

Authors Dostal, Kozubek, Kovar, Markopoulos, and Brzobohaty use exactly the Perron vector for fixing nodes computation in their generalized inverse algorithm [1].

In Fig. 1 overtaken from [1], we can see that the position of the fixing node in the rightmost side subdomain differs from the position where we naturally expect it should be (i.e. closer to the “center” of given subdomain). One can see that the highest value of the Perron

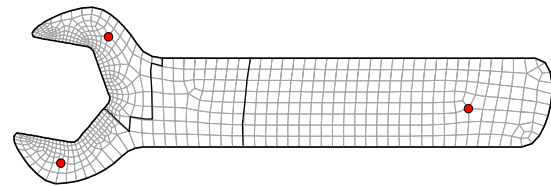


Fig. 1: Mesh with wrong position of fixing nodes.

vector arises at the vertex with the higher degree rather than in one with the mean value of the degree.

The last presented approach how to obtain a good approximation of the best choice of one-fixing node mentioned in this text is based on the eigenvectors of the Laplacian matrix.

The Laplacian matrix can be set up in dependence on the boundary conditions. As standard Laplacian matrix corresponds to problem with the Neumann boundary, it is more suitable to represent the problem that we are interesting in.

According to M. Fiedler theory [4], [5], the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix is used to graph partitioning [6], [7]. It is known, that the so-called Fiedler cut defines the cut in certain coordinate direction. The interesting thing is that in two-dimensional case, the eigenvector corresponding to the third smallest eigenvalue defines the cut in the second coordinate direction. Because the Fiedler cut is known to be somehow “optimal”, it is reasonable to expect a good approximation of one-fixing node near the crossing of both cuts.

For purposes of definition of a generalized eigenvector cut, the notation based on “Fiedler’s tree theorem” is used, especially notation of the characteristic vertex and the characteristic edge.

Using the other Fiedler’s results, the Fiedler cut is formed by the characteristic set (i.e. set of characteristic vertices or characteristic edges). In general, characteristic set can be defined for arbitrary eigenvector of the Laplacian matrix and for arbitrary type of graph.

Definition 4. *k-level cut:* Let $G = (V, E)$ be a graph on n vertices, labelled $1, 2, \dots, n$, with Laplacian matrix $\mathbf{L}(G)$. Let v_k be an eigenvector of $\mathbf{L}(G)$ associated with the k -th eigenvalue ($v_k[i]$ denotes i -th element of v_k).

The k -level cut is defined as a characteristic set, i.e. either set of characteristic vertices or set of characteristic edges, where:

- characteristic vertices are the vertices i that satisfy $v_k[i] = 0$,
- characteristic edges are the edges ij such that i and j are adjacent in G , $v_k[i] > 0$ and $v_k[j] < 0$ (or $v_k[i] < 0$, $v_k[j] > 0$).

Following this notation, the 2-level cut is exactly the Fiedler cut. As the spectrum of graph (or meshes respectively), respects the physical properties of given objects, i.e. the space-dimension, also the cross-eigenvector center has to be defined with respect to this characteristic (dimension of the mesh).

Definition 5. *Cross-eigenvector center:* Let $G = (V, E)$ be a graph on n vertices:

- For one-dimensional mesh, the cross-eigenvector center is the vertex, edge respectively, that lies on the 2-level cut.
- For two-dimensional mesh, the cross-eigenvector center is the vertex, edge or 2D element respectively, that lies on the crossing of the 2-level cut and the 3-level cut.
- For three-dimensional mesh, the cross-eigenvector center is the vertex, edge, face or 3D element respectively, that lies on the crossing of the 2-level cut, 3-level cut and 4-level cut.

In computational arithmetic, i.e. in presence of rounding errors, there is usually problem with zero identification. The cross-eigenvector center is always assigned to the vertex with the smallest value (in absolute value). The exact position of the cross-eigenvector center can differ no more than half edge, half element respectively.

In Section 5. there is presented the sketch of the proof that the cross-eigenvector center delivers the best choice of one-fixing node (instead of the other center-like points), which is its biggest advantage. In opposite to the capital vertex, the eigenvectors corresponding to the second, third, and fourth smallest eigenvalue of the Laplacian matrix are harder to compute than the eigenvector corresponding to the largest eigenvalue of the adjacency matrix, which is a disadvantage.

4. Experiments

The results of positioning of fixing nodes based on listed definitions are presented in this section. The criterion to measure quality of the approximation of one-fixing node is the condition number of the regular part \tilde{A}_{JJ} of the matrix \mathbf{A} of given problem Eq. (4) after permutation of the row and column corresponding to the fixing node at the end of the matrix.

Given examples are quite small (less than 121 vertices and 100 edges) because the finding of the best choice of one-fixing node according to Definition 1 yields to removing all vertices, computing the condition number of the regular part of the matrix (i.e. the largest and the smallest eigenvalue) for each removal,

and choosing the minimal one which is a huge time-consuming process.

In figures the following symbols appear:

- The best choice of one-fixing node satisfying Definition 1 is drawn as a circle \bigcirc .
- The graph centers are drawn as a square \blacksquare .
- The capital vertex satisfying Definition 3 is drawn as a triangle \triangle .
- The cross-eigenvector center satisfying Definition 5 is drawn as a star \star .

Let us have a look in the rightmost subdomain in Fig. 2. As the cross-eigenvector center (color cyan) almost the same vertex as the best choice of one-fixing node (color green) has been detected, meanwhile the capital vertex (color red) has been assigned to the vertex with higher degree which is far from the best choice of one-fixing node.

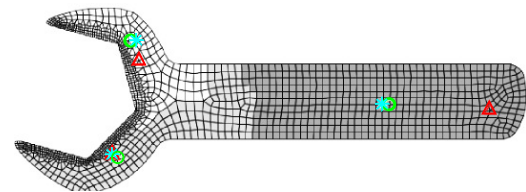


Fig. 2: Comparison of two approaches.

Let us briefly present the result of this phenomenon.

At first, let us have a look on the behaviour of center-like points of the meshes with regular elements, see Fig. 3(a). In two-dimensional space, the element corresponds to the graph face (region bounded by edges). Our examples consist of quadrilateral type elements only. As we can see from Fig. 3(a) and from the Tab. 1, almost all approaches deliver a good approximation of one-fixing node.

One can suppose that the behaviour of all center-like points is similar. But this holds only for meshes with uniform elements, or similarly, for meshes with inner vertices of the same degree. Let us have a look on the mesh with mixed elements (quadrilateral and triangle type) for further testing. This means that the degree of inner vertices differs from four to six.

For well understanding of the problem, several eigenvector of testing meshes are plotted (compare Fig. 3, Fig. 4). In Fig. 3(b), Fig. 4(b), the eigenvectors of the adjacency matrix corresponding to the highest eigenvalue are plotted. In Fig. 3(c), Fig. 4(c), the eigenvectors of the Laplacian matrix corresponding to the second smallest eigenvalue are plotted, and in Fig. 3(d), Fig. 4(d), the eigenvectors of the Laplacian matrix corresponding to the third smallest eigenvalue are plotted.

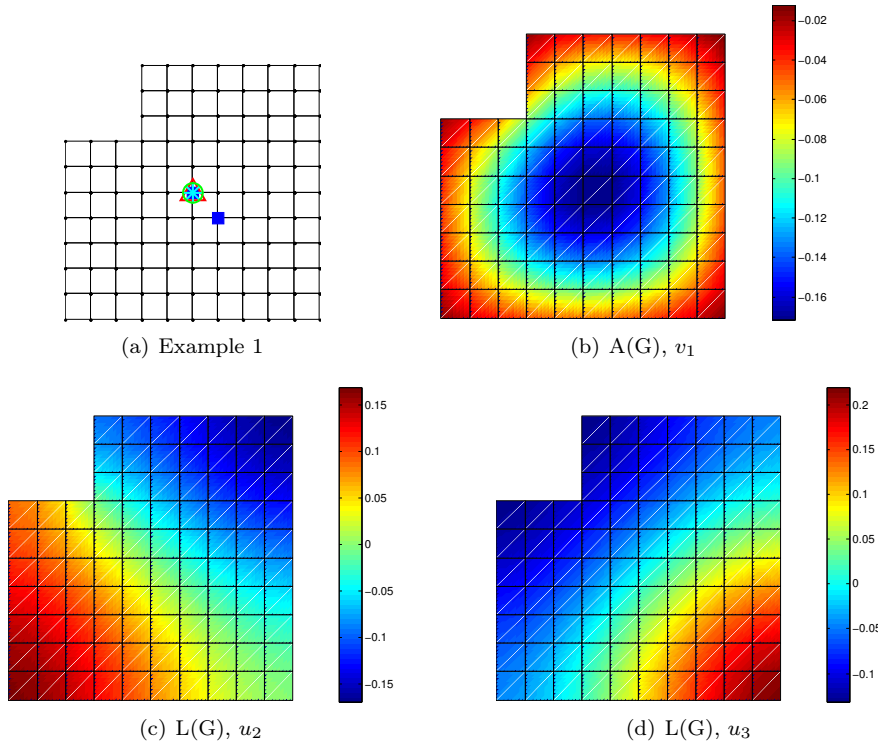


Fig. 3: Example 1 with plotted eigenvectors.

The testing mesh (Example 1) is similar to the Cartesian product of two paths $P_{11} \times P_{11}$ and also, the spectrum of this mesh is similar to the spectrum of Cartesian product $P_{11} \times P_{11}$. The mesh Example 1 can be understood as an approximate Cartesian product with deleted both edges and vertices. The missing left upper corner has no essential influence on the behaviour of the eigenvectors.

Another situation arises if the edges are added instead of deleted (see Fig. 4). The behaviour of the eigenvector of the adjacency matrix corresponding to the highest eigenvalue does not correspond to the classical behaviour of the spectrum of Cartesian products. Therefore, the capital vertex (based on the adjacency matrix) do not approximate the best choice of one-fixing node well.

As we can see in Fig. 4(c), Fig. 4(d), the eigenvectors of the Laplacian matrix are still indifferent to change of the structure of the mesh. The cross-eigenvector center still approximates the best choice of one-fixing node well.

For better comprehension, Tab. 1 is presented. The first column represents the name of the test example. The regular condition number of the matrix \mathbf{A} in the second column is computed by:

$$\overline{\text{cond}}(\mathbf{A}) = \frac{\lambda_{max}(\mathbf{A})}{\lambda_{min}(\mathbf{A})}, \tag{7}$$

where $\lambda_{max}(\mathbf{A})$ and $\lambda_{min}(\mathbf{A})$ correspond to the largest and to the smallest non-zero eigenvalue of \mathbf{A} . The regular condition number is used as a reference value, because it represents the smallest boundary to the condition number of the generalized inverse (the condition number of the generalized inverse can be never smaller than the condition number of the original matrix). The third column represents the condition number of the matrix \tilde{A}_{JJ} considered in Definition 1, i.e. the minimal possible condition number of the regular part \tilde{A}_{JJ} after removing the best choice of one-fixing node. The condition number of the regular part \tilde{A}_{JJ} after removing the graph center (without definition) is written in the fourth column.

If more vertices satisfy the definition, the vertex that causes the minimal condition number has been chosen. The fifth column represents the condition number of the matrix \tilde{A}_{JJ} when the capital vertex satisfying Definition 3 is removed. The last column represents the condition number of the matrix \tilde{A}_{JJ} when the cross-eigenvector center satisfying Definition 5 is removed.

We can see that choosing the capital vertex as the approximation of the best choice of one-fixing node instead the cross-eigenvector center debases the size of the condition number from 426.5 (best choice of one-fixing node, cross-eigenvector center) to 537.1, i.e. by 26 %.

Tab. 1: Approximation of the best choice of one-fixing node.

No.	$\overline{\text{cond}}(A)$ (regular)	$\kappa(\tilde{A}_{JJ})$ (1-fixing node)	$\text{cond}(\tilde{A}_{JJ})$ (graph center)	$\text{cond}(\tilde{A}_{JJ})$ (Adjacency m.)	$\text{cond}(\tilde{A}_{JJ})$ (Laplace m.)
1	107.7	379.8	379.8	379.8	379.8
2	99.2	426.5	426.5	537.1	426.5

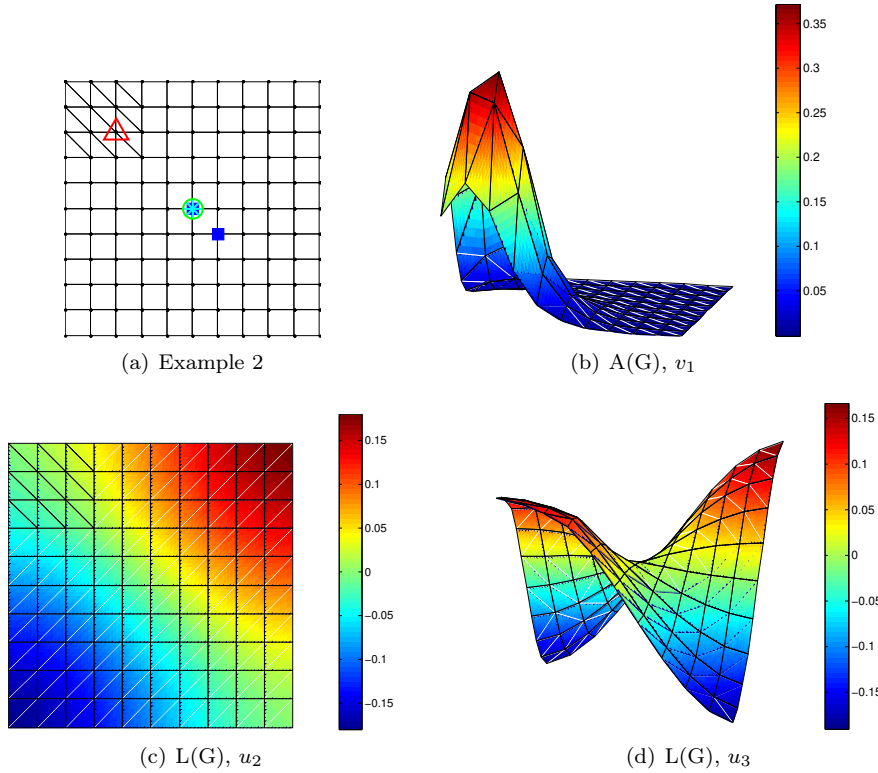


Fig. 4: Example 2 with plotted eigenvectors.

5. The Best Choice of One-Fixing Node

Based on the experiments, it is seemed that the best approximation of the best choice of one-fixing node is the cross-eigenvector center. In this section, the particular theorem is presented.

Theorem 6. *The best choice of one-fixing node: The cross-eigenvector center (according to Definition 5) is the best choice of one-fixing node (according to Definition 1).*

I.e. if we remove the row and column corresponding to the cross-eigenvector center from the original matrix \mathbf{A} , the remaining principal submatrix has the best condition number over all principal submatrices.

Remark that we restrict on the matrices with one-dimensional kernel, i.e. the resulting submatrix has deleted one row and one corresponding column. In

above definitions, we have worked with some general stiffness matrix \mathbf{A} . As we consider the Laplacian matrix of graph, we will use the symbol \mathbf{L} hereinafter. Symbols with \sim as $\tilde{\mathbf{L}}$, $\tilde{\lambda}$, etc. will further denote variables of corresponding reduced problem. If we could emphasize that the i -th row/column is removed, we assign the corresponding variables as $\tilde{\mathbf{L}}^i$, $\tilde{\lambda}^i$, etc.

The condition number is considered in the form:

$$\text{cond}(\tilde{\mathbf{L}}) = \frac{\tilde{\lambda}_{\max}(\tilde{\mathbf{L}})}{\tilde{\lambda}_{\min}(\tilde{\mathbf{L}})}. \tag{8}$$

From one of the variant of Cauchy-like interlacing theorems (see i.e. [8]), removing the fixing node does not change the $\tilde{\lambda}_{\max}$ so much as the $\tilde{\lambda}_{\min}$. Therefore, the minimization of the condition number corresponds to the maximization of the $\tilde{\lambda}_{\min}$. Theorem 6 can be paraphrased into following form:

Theorem 7. *Maximization of $\tilde{\lambda}_{\min}$: Removing the vertex k corresponding to the cross-eigenvector center*

in Laplacian matrix \mathbf{L} , the maximal value of $\tilde{\lambda}_{min}^*$ of $\tilde{\mathbf{L}}$ is obtained over all $\tilde{\lambda}_{min}^i$ of all principal submatrices $\tilde{\mathbf{L}}^i$, i.e., the condition number $\tilde{\mathbf{L}}^i$ is minimized over all i .

Finding the maximal value of $\tilde{\lambda}_{min}^*$ can be written:

$$\tilde{\lambda}_{min}^* = \max_{i=\{1,2,\dots,n\}} \tilde{\lambda}_{min}^i. \tag{9}$$

As the whole proof of this theorem consists in comprehensive analysis of all cases, we restrict on the main ideas only. The complete proof can be found in [2].

Proof. In our considerations, we come out from the Rayleigh's principle [9]. The smallest eigenvalue of the reduced matrix $\tilde{\mathbf{L}}$ can be computed as:

$$0 \neq \tilde{\lambda}_{min} = \min_{\|x\|=1} \frac{x^T \tilde{\mathbf{L}} x}{x^T x} = \min_{\|x\|=1} x^T \tilde{\mathbf{L}} x. \tag{10}$$

As we would like to find which vertex i from the original matrix \mathbf{L} should be removed to obtain the maximal value of $\tilde{\lambda}_{min}$, we have to rewrite the above equation using original matrix \mathbf{L} . Remark that the removing of the vertex i in original matrix \mathbf{L} corresponds to setting the corresponding component of vector x to zero, which can be written as:

$$\begin{aligned} \tilde{\lambda}_{min}^i &= \min_{\substack{\|x\|=1 \\ x_i=0}} x^T \mathbf{L} x = \\ &= \min_{\substack{\|x\|=1 \\ x_i=0 \sim x^T e_i = 0 \sim e_i^T x = 0}} x^T \mathbf{Q} \Lambda \mathbf{Q}^T x. \end{aligned} \tag{11}$$

In the last equation, we have substitute the Laplacian matrix by its spectral decomposition, where Λ is a diagonal matrix of eigenvalues of \mathbf{L} ordered in ascending order and \mathbf{Q} is the matrix, whose columns are eigenvectors of \mathbf{L} ordered accordingly.

After some manipulations (and substituting $y = \mathbf{Q}^T x$, $y^T = x^T \mathbf{Q}$ respectively) we get:

$$\begin{aligned} \tilde{\lambda}_{min}^i &= \min_{\substack{\|y\|=1 \\ r_i(\mathbf{Q})y=0}} y^T \Lambda y = \\ &= \min_{\substack{\|y\|=1 \\ r_i(\mathbf{Q})y=0}} \sum_{i=1}^n \lambda_i, \end{aligned} \tag{12}$$

where $r_i(\mathbf{Q})$ denotes the i -th row of matrix \mathbf{Q} (and it is not the eigenvector).

For given n we solve following system of equations:

$$\tilde{\lambda}_{min}^i = \min (\lambda_1 y_1^2 + \lambda_2 y_2^2 + \dots + \lambda_n y_n^2), \tag{13}$$

subject to:

$$y_1^2 + y_2^2 + \dots + y_n^2 = 1, \tag{14}$$

$$q_{i1}y_1 + q_{i2}y_2 + \dots + q_{in}y_n = 0. \tag{15}$$

Here, q_{ij} denotes the ij -th entry of matrix \mathbf{Q} , i.e. q_{i*} denotes again the i -th row of matrix \mathbf{Q} .

Thanks to Cauchy Interlacing Theorem [8]:

$$0 = \lambda_{1=min} \leq \tilde{\lambda}_{1=min} \leq \lambda_2 \leq \dots \leq \tilde{\lambda}_{n-1} \leq \lambda_n, \tag{16}$$

we get the upper bound for $\tilde{\lambda}_{min}^i \leq \lambda_2$.

Considering the particular example of regular rectangle mesh with known eigenvalues and eigenvectors, it can be shown, that we obtain exactly the upper bound for i such that $q_{i2} = 0$, [2]:

$$\tilde{\lambda}_{min}^i = \min (\lambda_1 y_1^2 + \lambda_2 y_2^2 + \dots + \lambda_n y_n^2) = \lambda_2. \tag{17}$$

Notice that in this notation, the q_{i2} corresponds to the i -th entry of the second smallest eigenvector of the Laplacian matrix which corresponds exactly to 2-level cut according to Definition 4.

6. Conclusion

In this paper, we have presented a modification in stabilization of generalised inverse algorithm used in [1]. Our modification consists in another choice of fixing node.

Several numerical experiments have been done to observe the influence of different choice of fixing nodes on the condition number of resulting matrix. Based on these experiments, it has been shown that using eigenvectors of the Laplacian matrix provides better choice of fixing node than using eigenvector of the adjacency matrix (used in [1]).

We have presented the main ideas of the analytical proof that the capital vertex (defined using eigenvectors of the Laplacian matrix) provides the best choice of one-fixing node. The whole proof is based on knowledge of eigenvalues and eigenvectors of the particular example of regular rectangle mesh and it can be found in [2].

References

[1] BRZOBOHATY, T., Z. DOSTAL, T. KOZUBEK, P. KOVAC and A. MARKOPOULOS. Cholesky decomposition with fixing nodes to stable computation of a generalized inverse of the stiffness matrix of a floating structure. *International Journal for Numerical Methods in Engineering*. 2011, vol. 88, iss. 5, pp. 493–509. ISSN 0029-5981. DOI: 10.1002/nme.3187.

[2] KABELIKOVA, P. *Implementation of Non-Overlapping Domain Decomposition Techniques for FETI Methods*. Ostrava, 2012. Dissertation thesis. VSB–Technical University of Ostrava. Supervisor Vit Vondrak.

- [3] KABELIKOVA, P. Graph centers used for stabilization of matrix factorizations. *Discussiones Mathematicae Graph Theory Journal*. 2010, vol. 30, iss. 2, pp. 249–259. ISSN 1234-3099.
- [4] FIEDLER, M. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*. 1973, vol. 23, iss. 98, pp. 298–305. ISSN 0011-4642.
- [5] FIEDLER, M. A property of eigenvectors of non-negative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*. 1975, vol. 25, iss. 100, pp. 619–633. ISSN 0011-4642.
- [6] BARNARD, S. T. and H. D. SIMON. A Fast Multilevel Implementation of Recursive Spectral Bisection for Partitioning Unstructured Problems. *Concurrency: Practice and Experience*. 1992, vol. 6, iss. 2, pp. 101–117. ISSN 1532-0634.
- [7] POTHEN, A., H. D. SIMON and K. LIOU. Partitioning Sparse Matrices with Eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*. 1990, vol. 11, iss. 3, pp. 430–452. ISSN 1095-7162. DOI: 10.1137/0611030.
- [8] HORN, R. A. and Ch. R. Johnson. *Matrix Analysis*. Cambridge: Cambridge University Press, 1985. ISBN 978-0-521-38632-6.
- [9] QUENELL, G. T. Eigenvalue comparisons in graph theory. *Pacific Journal of Mathematics*. 1996, vol. 176, iss. 2, pp. 619–633. ISSN 0030-8730.

About Authors

Pavla HRUSKOVA was born in Cesky Tesin as Pavla Kabelikova. She received her M.Sc. from Computer Science and Technology in 2006 and her Ph.D. from Computational and Applied mathematics in 2012, both at VSB–Technical University of Ostrava. Her research interests include graph theory with particular emphasis on the spectral graph theory, numerical mathematics and numerical modelling, and high performance computing. She has married in 2013 and she adopted surname Hruskova. At present, she works at the Department of Applied Mathematics of Faculty of Electrical Engineering and Computer Science of VSB–Technical University of Ostrava.