Fixing nodes of meshes from point of view of graph theory

Pavla Kabelíková

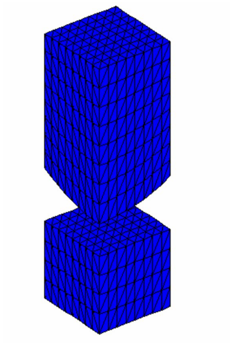
VŠB–Technical University of Ostrava

in cooperation with Prof. Clemens Brand

December 14, 2009

Outline

Numerical mathematics: Contact problem



Primal problem: $\min \frac{1}{2} u^T K u - u^T f, \quad B_1 u \leq c_1.$

Problem in computation

$$K = \begin{bmatrix} K_1 & O \\ O & K_2 \end{bmatrix}$$

$$K_1 \cdots SPS, \qquad K_1 u_1 = f_1, \qquad f_1 \in Im(K_1), \quad f_1 = K_1 y_1.$$

Problem in computation

$$K = \left[\begin{array}{cc} K_1 & O \\ O & K_2 \end{array} \right]$$

$$K_1 \cdots SPS, \qquad K_1 u_1 = f_1, \qquad f_1 \in Im(K_1), \quad f_1 = K_1 y_1.$$

- Evaluation of the action of generalized inverse of the SPS stiffness matrix K of floating subdomains:

$$K = KK^+K \qquad \Rightarrow \qquad u = K^+ f,$$

where K^+ is a one-condition generalised inverse.

- Solution:

$$Ku = KK^+ f = KK^+ Ky = Ky = f$$

Computation of the generalized inverse of SPS matrix

- Cholesky decomposition, LU decomposition
  - setting to zero the columns which correspond to zero pivots (rounding errors, the nonsingular part of A is ill-conditioned).
- Singular Value Decomposition (SVD)
  - good results, only for relativelly small matrices!
- Combination of Cholesky decomposition and SVD
  - quality of SVD + fastness of Cholesky decomposition (Farhat and Geradin, Papadrakakis and Fragakis).

Computation of the generalized inverse of SPS matrix

3. Combination of Cholesky decomposition and SVD
(Farhat and Geradin, Papadrakakis and Fragakis)

- Starting with Cholesky decomposition.
- In case of "suspected" pivot (zero pivot) starts SVD.

Problem: Zero pivot can arise at the beginning $\Rightarrow$ SVD on large matrices and ill-conditioned regular part of matrix A.

Modified Farhat-Gerardin Algorithm

Decomposition of the matrix $A \in R^{n\times n}$:

$$PAP^T = \begin{bmatrix} \widetilde{A}_{JJ} & \widetilde{A}_{JI} \\ \widetilde{A}_{IJ} & \widetilde{A}_{II} \end{bmatrix} = \begin{bmatrix} L_{JJ} & O \\ L_{IJ} & I \end{bmatrix} \begin{bmatrix} L_{JJ}^T & L_{IJ}^T \\ O & S \end{bmatrix},$$

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

Modified Farhat-Gerardin Algorithm

Then

$$A^+ = P^T \begin{bmatrix} L_{JJ}^{-T} & -L_{JJ}^{-T} L_{IJ}^T S^\dagger \\ O & S^\dagger \end{bmatrix} \begin{bmatrix} L_{JJ}^{-1} & O \\ -L_{IJ} L_{JJ}^{-1} & I \end{bmatrix} P,$$

- $S^\dagger$ denotes the Moore–Penrose generalized inverse.

Modified Farhat-Gerardin Algorithm

Finding P:

1. P_1:

$$P_1 A P_1^T = \begin{bmatrix} \widetilde{A}_{JJ} & \widetilde{A}_{JI} \\ \widetilde{A}_{IJ} & \widetilde{A}_{II} \end{bmatrix},$$

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

Modified Farhat-Gerardin Algorithm

Finding P:

1. P_1:

$$P_1 A P_1^T = \left[\begin{array}{cc} \widetilde{A}_{JJ} & \widetilde{A}_{JI} \\ \widetilde{A}_{IJ} & \widetilde{A}_{II} \end{array} \right],$$

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

2. P_2: reordering algorithm on $P_1 A P_1^T$ to get a permutation matrix P_2 which leaves the part $\widetilde{A}_{II}$ without changes and enables the sparse Cholesky factorization of $\widetilde{A}_{JJ}$.

Then $P = P_2 P_1$.

Detection of Fixing Nodes

Consider the mesh of the problem as an **unweighted non-oriented graph** and the matrix A as an Laplacian matrix of the original mesh.

Detection of Fixing Nodes

Consider the mesh of the problem as an **unweighted non-oriented graph** and the matrix A as an Laplacian matrix of the original mesh.

- Conditioning of A seems to be related with detection of 'fixing nodes' such that the related Dirichlett conditions make the structure as stiff as possible.
- Having the 'fixing nodes', we reorder the matrix A so that the rows corresponding to these nodes (degrees of freedom) are at the bottom.
- These rows represent the submatrix $\widetilde{A}_{II}$.

Detection of Fixing Nodes

Consider the mesh of the problem as an **unweighted non-oriented graph** and the matrix A as an Laplacian matrix of the original mesh.

- Conditioning of A seems to be related with detection of ‘fixing nodes’ such that the related Dirichlett conditions make the structure as stiff as possible.
- Having the ‘fixing nodes’, we reorder the matrix A so that the rows corresponding to these nodes (degrees of freedom) are at the bottom.
- These rows represent the submatrix $\widetilde{A}_{II}$.

Now, our problem is to **identify the fixing nodes** in the graph.

Definition of Fixing Nodes

Definition 1 Fixing nodes.
We define *i-fixing nodes* to be the set of *i* nodes that make the matrix of a given problem nonsingular and well conditioned, i.e. the removing of these *i* nodes makes the regular condition number of the matrix finite and sufficiently small.

The best choice of *i*-fixing nodes is then the set of fixing nodes, that make the numerical solution as stable as possible, i.e. the removing of these *i* nodes makes the regular condition number of the matrix of a given problem minimal.

Regular condition number κ is then computed as

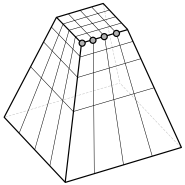
$$\kappa(\widetilde{A}_{JJ}) = \mathrm{cond}(\widetilde{A}_{JJ}) = \frac{\lambda_{max}(\widetilde{A}_{JJ})}{\lambda_{min}(\widetilde{A}_{JJ})}.$$

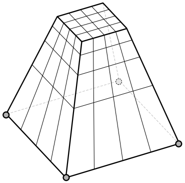
Definition of Fixing Node (one vertex)

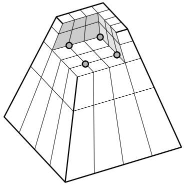
Definition 2 One fixing node.
Let $Ax = b$ be a system of linear equations such that A has one-dimensional kernel. We define the *best choice of fixing node* to be the node J that makes the numerical solution as stable as possible, i.e., the principal submatrix $\widetilde{A}_{jj}$, $j = J$, has the minimal condition number over all $\widetilde{A}_{jj}$.

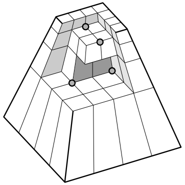
$$\kappa(\widetilde{A}_{JJ}) = \min_{\widetilde{A}_{jj},\ j=1,\cdots n} \mathrm{cond}(\widetilde{A}_{jj}) = \frac{\lambda_{max}(\widetilde{A}_{jj})}{\lambda_{min}(\widetilde{A}_{jj})},$$

Experiments


(a) $\kappa = 1.42 \times 10^{20}$


(b) $\kappa = 2622$

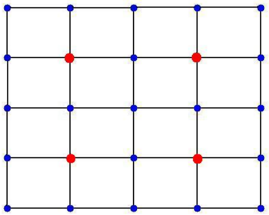

(c) $\kappa = 587$

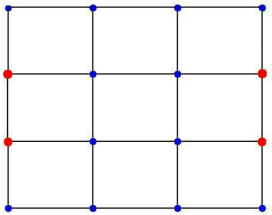

(d) $\kappa = 435$

k-Graph center

Let us consider the problem of finding the ‘fixing nodes’ as the problem of **finding a graph centers**.
A *k-graph center* as a set of k vertices:

$$\min_{\substack{C\subset V(G)\\|C|=k}} \max_{v\in V(G)} dist(C,v) = \min_{\substack{C\subset V(G)\\|C|=k}} \max_{v\in V(G)} \left(\min_{x\in C} dist(x,v)\right)$$

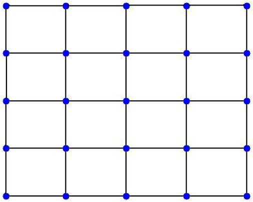


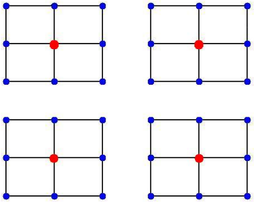


Speedup of computation $\rightsquigarrow$ **MatSol library**

Solution:

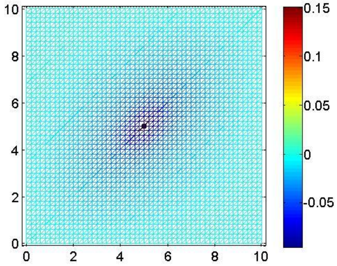
1. dividing the graph into k parts
   $\longrightarrow$ using METIS software,
2. finding one center in each part
   $\longrightarrow$ from the vertices that fit the basic definition choose the nearest vertex to the geometrical center.



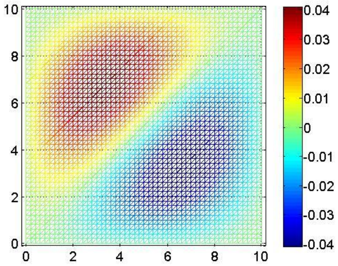


Spectral Approach

- The eigenvalues and the eigenvectors espetially express some characteristic of graphs.



(a) first eigenvector



(b) second eigenvector

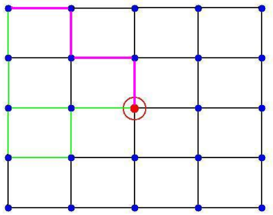
Spectral Approach

- We work with the (Perron) eigenvector corresponding to the largest eigenvalue of the adjacency matrix. The maximum entry of this eigenvector (in absolute value) corresponds to the fixing node.
- One of the methods suitable for computing this eigenvector is the Power method.

Spectral Approach

Lemma

Let A be the adjacency matrix of a given mesh and let $B = A^k$. Each element b_{ij} of B gives the number of distinct (i, j)-walks of length k in the mesh.



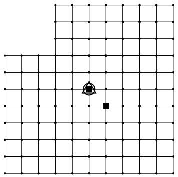
We call the node corresponding to the maximum entry in matrix B as *"eigenvector center"* ($1-$*eigenvector center*).

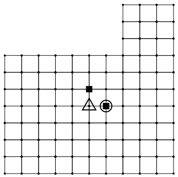
Experiments 2

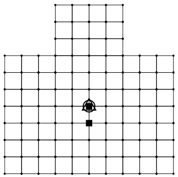
In following figures appear the following symbols.

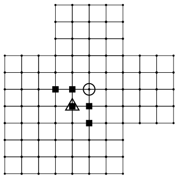
1. The fixing node is drawn as a circle $\bigcirc$.
2. The graph center is drawn as a square $\blacksquare$. When more vertices satisfy the definition, all such vertices are drawn.
3. The $1-$eigenvector center is drawn as a triangle $\triangle$.

Experiments 2

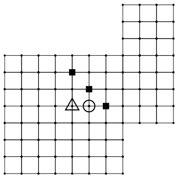

(a) Example 1

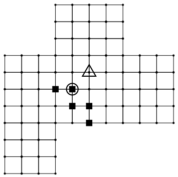

(b) Example 2

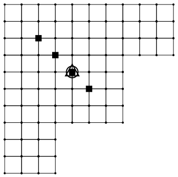

(c) Example 3

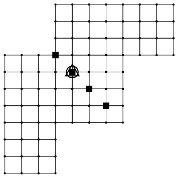

(d) Example 4

Experiments 2


(e) Example 5


(f) Example 6


(g) Example 7


(h) Example 8

No.	$\overline{\mathrm{cond}}(A)$	$\mathrm{cond}(\widetilde{A}_{JJ})$ (fixing n.)	$\mathrm{cond}(\widetilde{A}_{JJ})$ (eigenvec. c.)	$\mathrm{cond}(\widetilde{A}_{JJ})$ (last v.)
1	107.75	379.80	379.80	1228.4
2	130.00	341.24	345.26	1317.4
3	90.76	341.10	341.10	1111.1
4	98.44	308.78	317.04	995.3
5	142.69	314.74	341.05	1218.1
6	126.41	273.08	289.53	840.4
7	129.21	298.41	298.41	1124.8
8	144.63	270.21	270.21	1015.6

Table 1: Condition numbers.

Spectral Approach

Again, we do not require the optimal solution $\Rightarrow$ speedup of computation:

1. dividing the graph into k parts
   $\longrightarrow$ using METIS software,
2. finding one $1-$eigenvector center in each part.

Proof

We would like to prove theoretically:

The $1-$eigenvector center is the best choice of one fixing node.

I.e. if we remove the row and column corresponding to the $1-$eigenvector center from the original matrix A, the remaining principal submatrix has the the best condition number over all principal sumbatrices.
(In fact, we prescribe the Dirichlett condition to the given vertex)

Proof

For this purpose we consider several simplifications:

1. instead of in the maximum entry of the "largest" eigenvector of the adjacency matrix we consider the fixing node in the minimum entry (in absolute value) of the second "smallest" eigenvector of the Laplacian matrix.
2. As we consider the condition number in the form $\kappa(\widetilde{A}_{JJ}) = \lambda_{max}(\widetilde{A}_{JJ})/\lambda_{min}(\widetilde{A}_{JJ})$, we suppose, that removing the fixing node doesn't change the λ_{max} so much as the λ_{min}. **Thus, minimalization of the condition number corresponds to the maximalization of the λ_{min}.**
3. We consider eigenvectors in the normed form, e.g. $\|v_i\| = 1$.

Proof

From the Rayleigh quotient, the smallest eigenvalue can be computed as

$$\widetilde{\lambda}_{min} = \min_{\|x\|=1} x^T \widetilde{L} x = \min_{\substack{\|x\|=1 \\ x_i=0}} x^T L x$$

$$\widetilde{\lambda}_{min} \quad = \quad \min_{\substack{\|x\|=1 \\ e_i^T x=0}} x^T Q \Lambda Q^T x = \min_{\substack{\|y\|=1 \\ e_i^T Qy=0}} y^T \Lambda y = \min_{\substack{\|y\|=1 \\ y^T q_i=0}} y^T \Lambda y \quad (1)$$

The minimization problem can be then computed, e.g., in sense of Lagrange multiplires:

$$\min_y \left[y^T \Lambda y - s y^T q_i - t(y^T y - 1) \right] \quad (2)$$

Proof

By computing gradient we get

$$y = \tfrac{s}{2}(\Lambda - tI)^{-1} q_i, s \neq 0 \quad \text{and} \quad t = y^T \Lambda y = \widetilde{\lambda}_1$$

After several operations we get

$$q_i^T (\Lambda - tI)^{-1} q_i = 0$$

$$\sum_{k=1}^{n} q_{ik}^2 \frac{1}{\lambda_k - t} = 0 \tag{3}$$

In the equation (3), the λ_k are the eigenvalues of the original matrix L without removing fixing nodes, thus the smalest eigenvalue is zero $(0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n)$.

Proof

The equation (3) can be thus written as

$$-\frac{q_{i1}^2}{t}+\frac{q_{i2}^2}{\lambda_2-t}+\cdots+\frac{q_{in}^2}{\lambda_n-t}=0 \tag{4}$$

$$-\frac{1}{nt}+\frac{q_{i2}^2}{\lambda_2-t} \quad = \quad -\left(\sum_{k=3}^{n} q_{ik}^2\frac{1}{\lambda_k-t}\right) \tag{5}$$

The maximal value of the smallest eigenvalue $\widetilde{\lambda}_1$ of the reduced problem is obtainted in the vertex i, in which the entry of the second smallest eigenvector of the Laplacian matrix L is minimal (in absolute value, i.e. closest to zero).

Eigenvalues and eigenvectors of the path P_n

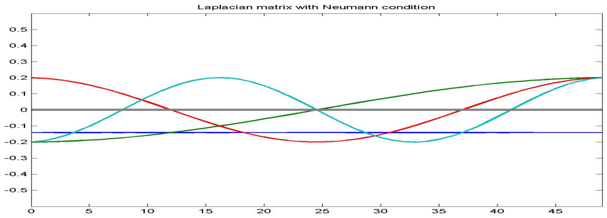
Laplacian matrix with Neumann boundary conditions:

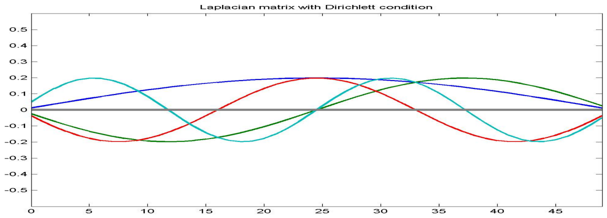
- $\lambda_k = 2 - 2cos(\frac{\pi k}{n})$
- $v_k(j) = cos(\frac{\pi k}{n}(j - \frac{1}{2}))$

Laplacian matrix with Dirichlett boundary conditions:

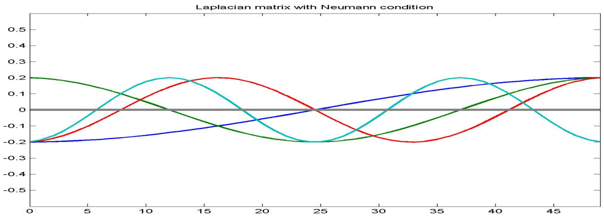
- $\lambda = 2 - 2cos(\frac{\pi k}{n})$
- $v_k(j) = cos(\frac{\pi kj}{n})$

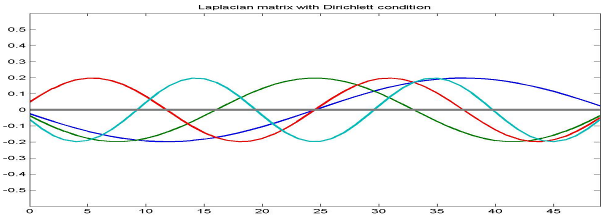
Eigenvectors of the path



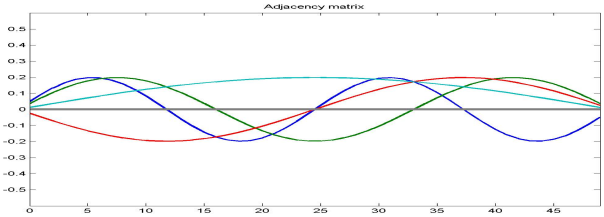


Eigenvectors of the path





Eigenvectors of the path



Eigenvalues and eigenvectors of Cartesian product $P_n \Box P_n$

$$L_{P_n \Box P_n} = L_{P_n} \otimes I_{P_n} + I_{P_n} \otimes L_{P_n}$$

Laplacian matrix with Neumann boundary conditions:

- The eigenvalues are all possible sums of $\lambda_i + \lambda_j$:

$$\lambda_{ij} = 4 - 2cos(\frac{\pi i}{n}) - 2cos(\frac{\pi j}{n}), \quad \forall i, j.$$

- The eigenvectors are Kronecker products of eigenvectors v_i, v_j:

$$v_k(ij) = cos(\frac{\pi k}{n}(i - \frac{1}{2})) \cdot cos(\frac{\pi k}{n}(j - \frac{1}{2})), \quad \forall i, j.$$

Eigenvectors of Cartesian product $P_n \square P_n$

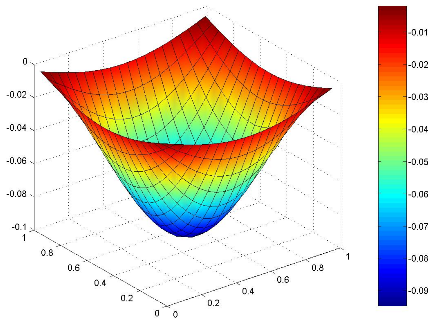
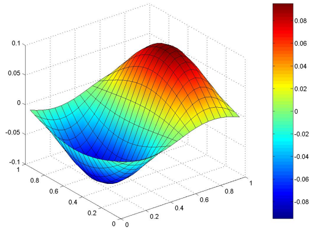
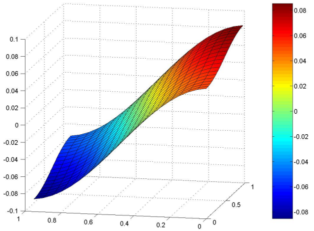


Figure: Adjacency matrix, 1st largest eigenvector

Eigenvectors of Cartesian product $P_n \square P_n$

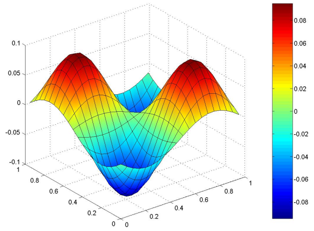


(a) Adjacency m., 2nd largest

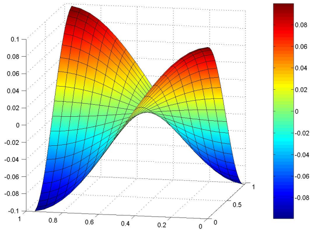


(b) Laplacian m., 2nd smallest

Eigenvectors of Cartesian product $P_n \Box P_n$



(c) Adjacency m., 4st largest



(d) Laplacian m., 4st smallest

Eigenvectors of Cartesian product $P_n \square P_n$

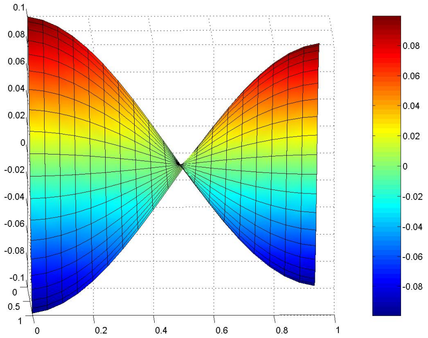


Figure: Laplacian matrix, 4th smallest eigenvector

Thank you for your attention.