Discussiones Mathematicae
Graph Theory 30 (2010) 349–359

GRAPH CENTERS USED FOR STABILIZATION OF MATRIX FACTORIZATIONS

Pavla Kabelíková

Department of Applied Mathematics
FEI, VSB – Technical University of Ostrava
17. listopadu 15, 708 33, Ostrava-Poruba, Czech Republic

e-mail: pavla.kabelikova@vsb.cz

Abstract

Systems of consistent linear equations with symmetric positive semidefinite matrices arise naturally while solving many scientific and engineering problems. In case of a "floating" static structure, the boundary conditions are not sufficient to prevent its rigid body motions.

Traditional solvers based on Cholesky decomposition can be adapted to these systems by recognition of zero rows or columns and also by setting up a well conditioned regular submatrix of the problem that is used for implementation of a generalised inverse. Conditioning such a submatrix seems to be related with detection of so called fixing nodes such that the related boundary conditions make the structure as stiff as possible. We can consider the matrix of the problem as an unweighted non-oriented graph. Now we search for nodes that stabilize the solution well-fixing nodes (such nodes are sufficiently far away from each other and are not placed near any straight line). The set of such nodes corresponds to one type of graph center.

Keywords: FETI, parallel computing, generalised inverse, graph center.

2010 Mathematics Subject Classification: 05C12, 05C50, 90C35.

1. Numerical Motivation

Systems of consistent linear equations with symmetric positive semidefinite (SPS) matrices arise naturally in the solution of many scientific and engineering problems. A typical example is the stress analysis of a "floating" static structure whose essential boundary conditions are not sufficient to prevent its rigid body motions [5, 12, 13, 14].

The consistent systems with a semidefinite matrix A can be solved either by an iterative method, such as the preconditioned conjugate gradient method [1], whose performance depends on the distribution of the spectrum of A, or by a direct method, typically based on a decomposition, whose performance depends on the sparsity pattern of A. Assuming exact arithmetic, it is rather easy to adapt standard direct methods for the solution of systems with positive definite matrices, such as the Cholesky decomposition, to the solution of systems with only positive semidefinite matrix [8]. The only modification comprises assigning zero to the columns which correspond to zero pivots. However, in agreement with the theoretical results of Pan [11], it turns out that it is very difficult to recognize the positions of such pivots in the presence of rounding errors when the nonsingular part of A is ill-conditioned. Moreover, even if the zero pivots are recognized, it turns out that the ill-conditioning of the nonsingular submatrix defined by the nonzero pivots can have a devastating effect on the precision of the solution.

1.1. Numerical solution

For more information about generalized inverse and Cholesky decomposition in exact arithmetic we refer to the paper [2]. Here, we would like to present only a simplified form of the numerical solution with regard to future needs.

As we consider contact problems, one of the method solving this type of problems is Finite Element Tearing and Interconnecting (FETI) method and its modifications. For background see [6], Total-FETI modification is presented in [4]. Our primal problem is

$$\min\left(\frac{1}{2}u^T K u - u^T f\right), \quad B_I u \le c_I, \quad B_E u = 0, \tag{1}$$

where K is a stiffness matrix of the given contact problem, f is a vector of forces, u is a vector of solution and matrices B_I, B_E represent boundary conditions.

Let us simplify the problem to solve $Ax = b$, thus $x = A^+b$. A is a symmetric positive semidefinite matrix of order n and $b \in \mathrm{ImA}$, such that the solution x exists. A^+ denotes the generalised inverse matrix. We are particularly interested in the case when the problem is solved many times with varying right hand sides. We shall assume that the sparsity pattern of A enables its effective triangular decomposition $A = LL^T$. The method of evaluation of the factor L is known as the *Cholesky factorization*:

$$PAP^T = \begin{bmatrix} \widetilde{A}_{JJ} & \widetilde{A}_{JI} \\ \widetilde{A}_{IJ} & \widetilde{A}_{II} \end{bmatrix} = \begin{bmatrix} L_{JJ} & O \\ L_{IJ} & Id \end{bmatrix} \begin{bmatrix} L_{JJ}^T & L_{IJ}^T \\ O & S \end{bmatrix}, \tag{2}$$

where $L_{JJ} \in R^{J\times J}$ is a lower factor of the Cholesky factorization of $\widetilde{A}_{JJ}$, $L_{IJ} \in R^{I\times J}$, $L_{IJ} = \widetilde{A}_{IJ}L_{JJ}^{-T}$, $S \in R^{I\times I}$ is a singular matrix of the defect d, $Id \in R^{I\times I}$ is the identity matrix. Indices I, J satisfy $I + J = n$, I is very much smaller than n. Finally, P is a permutation matrix which corresponds to both preserving sparsity and fixing nodes reordering.

Then

$$A^+ = P^T \begin{bmatrix} L_{JJ}^{-T} & -L_{JJ}^{-T}L_{IJ}^T S^\dagger \\ O & S^\dagger \end{bmatrix} \begin{bmatrix} L_{JJ}^{-1} & O \\ -L_{IJ}L_{JJ}^{-1} & Id \end{bmatrix} P, \tag{3}$$

where $S^\dagger$ denotes the (Moore-Penrose) generalized inverse. As we consider a symmetric matrix A, the matrix S is also trivially symmetric. Thus, $S^\dagger$ can be computed by the symmetric form of the Singular value decomposition of matrix S, $S^\dagger = V\Sigma^\dagger V^T$, where $V \in R^{I\times I}$ is orthogonal matrix, $VV^T = Id$, $\Sigma^\dagger = \mathrm{diag}\{\sigma_1^{-1}, \ldots, \sigma_{I-d}^{-1}, 0, \ldots, 0\} \in R^{I\times I}$ and $\sigma_1 \geq \cdots \geq \sigma_{I-d} > \sigma_{I-d+1} = \cdots = \sigma_I = 0$ are singular values of S.

2. Graph Theory

Let us now consider the matrix A of the problem as an unweighted non-oriented graph. In this paper, the the adjacency matrix of the original mesh is used. The adjacency matrix A of a finite unweighted non-oriented graph without loops or multiple edges on n vertices is the $n \times n$ matrix where the nondiagonal entry a_{ij} is the number of edges from vertex i to vertex j, and the diagonal entry a_{ii} is equal zero.

Conditioning of a matrix is strongly related to computational demandingness of an inverse or generalised inverse matrix. Conditioning of A seems

to be related with detection of so called "fixing nodes" such that the related boundary conditions make the structure as stiff as possible. Once we have the "fixing nodes", the matrix A is reordered so that the rows corresponding to these nodes are at the bottom of the matrix.

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.

From mechanical interpretation, the fixing nodes cannot lay on a straight line and therefore, they are at least three.

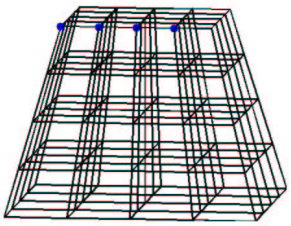
In notion of Equation (2), the permutation martix P reorders the matrix A so that the rows corresponding to fixing nodes are at the bottom of matrix A. Thus, the singular matrix $\widetilde{A}_{II}$ corresponds to i-fixing nodes. Regular condition number κ is then computed as $\kappa = \mathrm{cond}(\widetilde{A}_{JJ}) = |\lambda_{max}|/|\lambda_{min}|$, where λ_{max} and λ_{min} correspond to the largest and to the nonzero smallest eigenvalue of regular matrix $\widetilde{A}_{JJ}$, respectively.

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

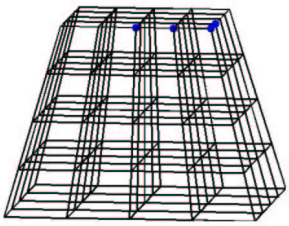
2.1. Identifying the fixing nodes

For testing purposes, we have chosen an object called "pyramid", see Figure 1. It is covered by three-dimensional mesh with five nodes in each direction. As we can see, the structure and the mesh are irregular (edges have not the same geometric length) with refinement on the top.

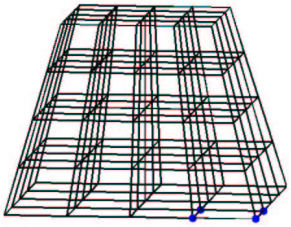
We have tested various placement of the fixing nodes in the mesh. The testing criterion was the condition number κ. The smaller condition number the faster computation of the generalised inverse matrix will be. We have decided to present the test with 4-fixing nodes. As we have a three-dimensional problem, the minimum number of fixing nodes is three to prevent rigid body motions in all three directions. Natural requirement is that the nodes are not near any straight line. When using four nodes, it turns out that nodes should not be coplanar.



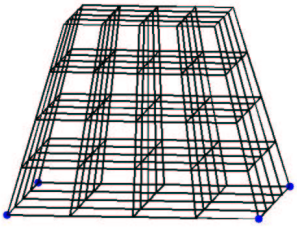
(a) $\kappa = 1.42 \times 10^{20}$



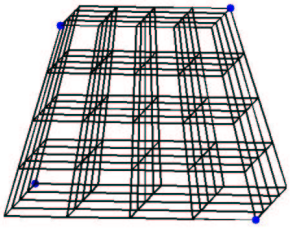
(b) $\kappa = 129471$



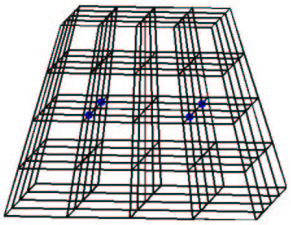
(c) $\kappa = 27015$



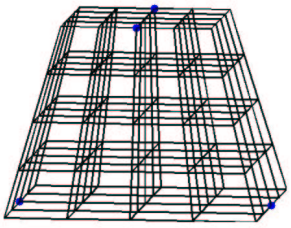
(d) $\kappa = 2622$



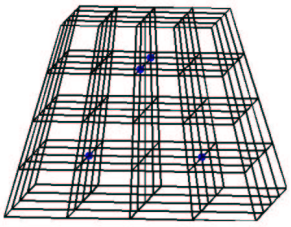
(e) $\kappa = 1140$



(f) $\kappa = 587$



(g) $\kappa = 527$



(h) $\kappa = 435$, the best choice

Figure 1. Pyramid: Finding the fixing nodes.

The upper bound on the number of fixing nodes is not exactly limited. As we increase the number of fixing nodes the condition number of matrix A decreases but we have to consider that the computational complexity of a generalised inverse increases. It turned out, that the 4-fixing nodes are sufficiently enough as long as we choose a proper set of nodes.

As we can see in Figure 1, the best choice of 4-fixing nodes is obtained when we place the 4-fixing nodes not coplanar in the interior of the object. This result lead to the idea to consider the problem or its generalization of finding fixing nodes as the problem of *finding graph centers*.

3. Graph Centers

In this section we present the definitions of graph centers related to our problem.

The definition of graph center for one vertex is the follows.

Definition 2. Let us define the *graph center* of a graph G as a vertex x, for which

$$\min_{x\in V(G)} \max_{v\in V(G)} \operatorname{dist}(x,v), \tag{4}$$

where $V(G)$ is the vertex set of a graph G, $\operatorname{dist}(x,v)$ is the distance between vertices x and v (length of the shortest path between these vertices).

The path in a graph G is a sequence of vertices $(v_1, v_2, \ldots, v_k)$, $v_1, v_2, \ldots, v_k \in V(G)$, such that all edges $v_i v_{i+1}$ are in the graph G for all $i = 1, 2, \ldots, k-1$. The shortest path between vertices $v_1 v_k$ is such a path, that the number of edges between the vertices $v_1 v_k$ is minimal.

There are several ways how to generalize the notion of a graph center. One is the following.

Definition 3. Let us define the *graph k-center* of a graph G as a set C of k vertices, for which

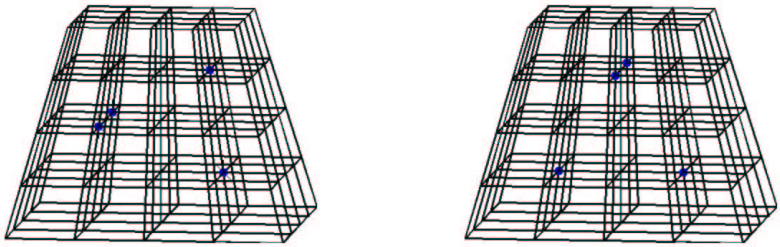
$$\min_{\substack{C\subset V(G)\\ |C|=k}} \max_{v\in V(G)} \operatorname{dist}(C,v) = \min_{\substack{C\subset V(G)\\ |C|=k}} \max_{v\in V(G)} \left(\min_{x\in C} \operatorname{dist}(x,v) \right), \tag{5}$$

where k is number of number of vertices in the center C, $V(G)$ is the vertex set of a graph G, $\mathrm{dist}(x, v)$ is the distance between vertices x and v (length of the shortest path between these vertices).

We have tested if the graph k-center satisfy the requirement on the best choice of k-fixing nodes in sense of Definition 1. When we tested object covered by regular mesh (all edges of the same geometrical length), graph k-center satisfied the best choice of k-fixing nodes, i.e., the regular condition number was minimal.

Remark 4. For regular meshes, we can consider the best choice of k-fixing nodes as graph k-center.

Problem arises, when the object is covered by an irregular mesh. In general, there could be more k-sets of vertices that satisfy Definition 3 but not all of them satisfy Definition 1.



(a) Suboptimal output: $\kappa = 492$ (b) Optimal output: $\kappa = 435$

Figure 2. Pyramid: suboptimal and optimal output.

In Figure 2, there are two different sets of four nodes that both satisfy Definition 3 but give a different condition number. The left figure shows output of our software that satisfy Definition 3 and the condition number is equal to 492. Experimentally we have found the optimal output depicted in the right figure, where the condition number is equal to 435 and that also satisfies the Definition 3. We can see that the condition number of the problem depends not only on the graph topology but also on the geometry of the mesh.

These results lead to the idea of a weightened graph, or to the idea of finding graph k-center in weightened graph, respectively. We suggest the following definition of k-center in weightened graph.

Definition 5. Let us define the *weightened graph k-center* of a graph G as a set C of k vertices, for which

$$\min_{\substack{C\subset V(G)\\|C|=k}} \max_{v\in V(G)} \operatorname{dist}_{\mathrm{w}}(C,v) = \min_{\substack{C\subset V(G)\\|C|=k}} \max_{v\in V(G)} \left(\min_{x\in C} \operatorname{dist}_{\mathrm{w}}(x,v)\right), \tag{6}$$

where k is the number of vertices in the center set C, $V(G)$ is the vertex set of a graph G, $\operatorname{dist}_{\mathrm{w}}(x,v)$ is the weightened distance between vertices x and v (length of the shortest edges-weightened path between those vertices).

$$\operatorname{dist}_{\mathrm{w}}(x,v) = \min_{P_{xv}\subset V(G)} \sum_{i=1}^{|P_{xv}|} w(e_i), \tag{7}$$

where P_{xv} is a path from x to v, $|P_{xv}|$ denotes the length of path P_{xv}, i.e., the number of edges that the path contains, $\{e_1, e_2, \ldots, e_{|P_{xv}|}\}$ are the edges on the path P_{xv}, $w(e_i)$ is the weight of edge e_i.

Now the problem is to find the weight function $w : E \to \mathcal{R}$, that assigns to each edge some real number characterizing its weight. This problem is not solved yet, because the weights of edges turned out to be dependent on the type of problem (two- or three-dimensional, type of mesh elements), on the mesure of refinement etc. Howeever, we suppose the weights should start at one and increase/decrease with very small difereces (e.g. logarithmical measure of geometrical distance). This idea corresponds to experimental results. These indicate that the graph topology is more significant than the geometrical distance.

3.1. Speedup of the computation

As we are interested in solving real problems with large number of variables, our main task is to find the fixing nodes fast. The algorithmical complexity of finding the best choice of fixing nodes should be in $O(n)$ or $O(n^2)$ complexity with massive usage of parallelism.

We shall point out that we do not strictly require an optimal solution. The idea of fixing nodes itself is very strong because solving systems with positive semidefinite matrix were not solved before in satisfactory time.

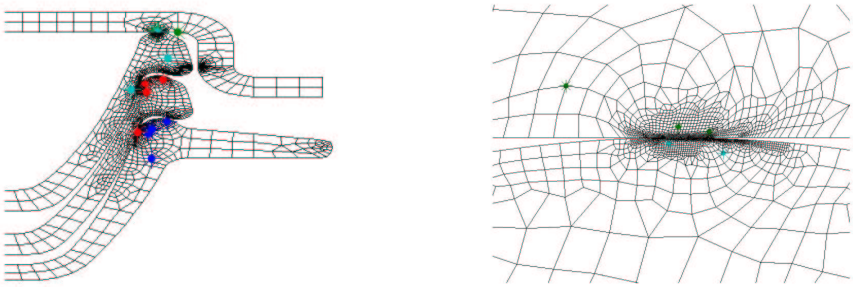
Thus, we do not need to obtain the best choice of fixing nodes, some sub-optimal solution obtained in a short time is sufficient.

In practical computations, the following algorithm consisting of two steps can be used.

1. Dividing the graph into k parts using some proper graph/mesh partitioning software (for example METIS, see [10]),
2. finding one graph center in each part.

We find one graph center in each part in the sense of Definition 2. From the vertices that satisfy the Definition 2 we choose the nearest vertex to the geometrical center.

The output of our algorithms of the given Example is shown in Figure 2(a). The output of our algorithms on a practical problem is shown in Figure 3. There is one symmetric part of a two-dimensional symmetric problem of mechanical mining supports consist of four components. In each part we have found four fixing nodes.



(a) (b)

Figure 3. Mining supports: four fixing nodes in each part.

Latest experiments show that the spectral theory is very powerful tool for finding fixing nodes. Especially, we use the (Perron) eigenvector corresponding to the largest eigenvalue of the adjacency matrix A. The maximum entry of this eigenvector (in absolute value) corresponds to one fixing node. Finding the eigenvector of a sparse symmetric adjacency matrix using some iterative method such as power method or Lanczos method is very fast comparing to the standard graphs methods for computing graph center. For more informations see [9].

4. Conclusion

In this paper we present the problem from numerical mathematics that turned out to be solved by means of graph theory. We have translated the problem of finding fixing nodes for purposes of computation of generalised inverse to the problem of finding graph center and we have suggested some definitions of graph k-center that satisfy the problem well. Also, we have presented the need of fast computation and we have introduced posibilitties how to achieve it as we do not require an optimal solution.

The future work in this area will consists of finding fast ways to find the fixing nodes with focus on spectral theory.

Acknowledgment

This work is supported by MSM 6198910027.

References

[1] O. Axelsson, Iterative Solution Methods (Cambridge University Press: Cambridge, 1994).

[2] T. Brzobohatý, Z. Dostál, T. Kozubek and A. Markopoulos, *Combining Cholesky decomposition with SVD to stable evaluation of a generalised inverse of the stiffness matrix of a floating structure*, submited, 2009.

[3] Z. Dostál, D. Horák, R. Kučera, V. Vondrák, J. Haslinger, J. Dobiáš and S. Pták, *FETI based algorithms for contact problems: scalability, large displacements and 3D Coulomb friction*, Computer Methods in Applied Mechanics and Engineering **194** (2005) 395–409.

[4] Z. Dostál, D. Horák and R. Kučera, *Total FETI — an easier implementable variant of the FETI method for numerical solution of elliptic PDE*, Communications in Numerical Methods in Engineering **22** (2006) 1155–1162.

[5] C. Farhat and M. Géradin, *On the general solution by a direct method of a large scale singular system of linear equations: application to the analysis of floating structures*, Int. J. Numer. Meth. Engng **41** (1998) 675–696.

[6] C. Farhat and F.-X. Roux, *A method of finite element tearing and interconnecting and its parallel solution algorithm*, Int. J. Numer. Meth. Engng **32** (1991) 1205–1227.

[7] Ch. Godsil and G. Royle, Algebraic Graph Theory (Springer-Verlag, New York, ISBN 0-387-95241-1, 2001).

[8] G.H. Golub and C.F. Van Loan, Matrix Computations (2nd ed.) (John Hopkins University Press, Baltimore, 1989).

[9] P. Kabelíková, T. Kozubek, A. Markopoulos and T. Brzobohatý, *Effective Algorithms for Implementation of FETI-Based Domain Decomposition Methods*, submited, 2009.

[10] G. Karypis and V. Kumar, *METIS manual Version* 4.0, [online] http://glaros.dtc.umn.edu/gkhome/views/metis, University of Minnesota, 1998.

[11] C.T. Pan, *On the existence and factorization of rank-revealing LU factorizations*, Linear Algebra and Its Applications **316** (2000) 199–222.

[12] M. Papadrakakis and Y. Fragakis, *An integrated geometric-algebraic method for solving semi-definite problems in structural mechanics*, Computer Methods in Applied Mechanics and Engineering **190** (2001) 6513–6532.

[13] K.C. Park and C.A. Felipa, *The construction of free-free flexibility matrices for multilevel structural analysis*, Int. J. Numer. Meth. Engng **47** (2000) 395–418.

[14] K.C. Park, C.A. Felipa and U.A. Gumaste, *A localized version of the method of Lagrange multipliers*, Computational Mechanics **24** (2000) 476-490.

Received 31 December 2008
Revised 30 Septemebr 2009
Accepted 9 November 2009