Francisco Pedroche; Miguel Rebollo; Carlos Carrascosa; Alberto Palomares
On some properties of the Laplacian matrix revealed by the RCM algorithm

Czechoslovak Mathematical Journal, Vol. 66 (2016), No. 3, 603–620

Persistent URL: http://dml.cz/dmlcz/145860

Czechoslovak Mathematical Journal, 66 (141) (2016), 603–620

ON SOME PROPERTIES OF THE LAPLACIAN MATRIX REVEALED BY THE RCM ALGORITHM

Francisco Pedroche, Miguel Rebollo, Carlos Carrascosa, Alberto Palomares, València

(Received June 27, 2015)

Dedicated to the memory of Miroslav Fiedler

Abstract. In this paper we present some theoretical results about the irreducibility of the Laplacian matrix ordered by the Reverse Cuthill-McKee (RCM) algorithm. We consider undirected graphs with no loops consisting of some connected components. RCM is a well-known scheme for numbering the nodes of a network in such a way that the corresponding adjacency matrix has a narrow bandwidth. Inspired by some properties of the eigenvectors of a Laplacian matrix, we derive some properties based on row sums of a Laplacian matrix that was reordered by the RCM algorithm. One of the theoretical results serves as a basis for writing an easy MATLAB code to detect connected components, by using the function "symrcm" of MATLAB. Some examples illustrate the theoretical results.

Keywords: ordering algorithm; reverse Cuthill-McKee algorithm; graph partitioning; Laplacian matrix

MSC 2010: 15B36, 05C50

1. Introduction

When dealing with a problem related to consensus in the framework of complex networks we noticed that a very easy MATLAB code allowed to detect connected components [25]. We were mixing concepts from Graph Theory (the Laplacian) and from Numerical Linear Algebra (the RCM algorithm). We thought it was worth getting a deep insight into the theoretical properties that these two concepts spread

The research has been supported by Spanish DGI grant MTM2010-18674, Consolider Ingenio CSD2007-00022, PROMETEO 2008/051, OVAMAH TIN2009-13839-C03-01, and PAID-06-11-2084.

on the table when we combine them. Since the proofs of the properties were beyond of the left-to-the-reader problems of typical postgraduate textbooks we considered it was interesting to write these proofs and to organize the theoretical material involved. As a result, we present in this paper some results about the irreducibility of the Laplacian matrix when it was reordered by the RCM algorithm. We also note that as a byproduct of one of these results, one can write a simple MATLAB code to detect connected components in graphs.

It is important to recall here that the standard method to detect connected components in networks is the *breadth first search* (BFS), see, e.g., [15], [13]. Note that our aim is not to offer a method faster than BFS, since RCM is a version of BFS, and moreover, we are using the implementation of RCM made by MATLAB. The BFS method can be implemented in running time of $O(n + m)$, n being the number of nodes and m the number of edges of the network. For sparse networks this means $O(n)$ but for dense networks we can have the case $m = O(n^2)$. BFS visits first the nodes that are closer to the initial node and lists all the neighbors. On the contrary, the method called the *depth first search* (DFS), or *backtracking*, visits first the nodes that are at a long distance from the initial node going as deep as possible. For the standard algorithm for the method DFS one usually refers to the seminal paper of Tarjan [29]. DFS can be implemented with a time complexity of $O(m)$, see [15] for details. Another variant of the BFS algorithm is the Reverse Cuthill-McKee algorithm (RCM). This method is based on the Cuthill-McKee algorithm, that was originally devised to reduce the bandwidth of symmetric adjacency matrices, see [4]. RCM can be implemented with a time complexity $O(q_{\max} m)$ where $q_{\max}$ is the maximum degree of any node, see [11]. Note that for sparse matrices we have that RCM works with time complexity $O(n)$. RCM operates over a matrix A and returns a permutation vector such that one can construct a symmetric permutation PAP^{T} that has a bandwidth smaller than the original one. It is known, see [4], that when some connected components are presented the RCM method leads to a block diagonal matrix similar to A. However, the RCM method is not commonly used to detect components.

The structure of the paper is the following. In Section 2 we recall some definitions from graph theory and matrix analysis. In Section 3 we describe the RCM algorithm and related work. In Section 4 we give some new theoretical results about the irreducibility of the Laplacian matrix; our contributions are Theorem 4.1 and Theorem 4.2. In Section 5 we derive some properties of row sums of a Laplacian matrix that was reordered by the RCM method; our contributions are Lemma 5.1 and Proposition 5.1. In Section 6 we mention a method to detect connected components by combining the Laplacian matrix and the RCM method. In Section 7 we give some conclusions.

2. Definitions and known results

Let $G = (V, E)$ be a graph, with $V = \{v_1, v_2, \ldots, v_n\}$ a nonempty set of n vertices (or nodes) and E a set of m edges. Each edge is defined by the pair (v_i, v_j), where $v_i, v_j \in V$. The adjacency matrix of the graph G is $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ such that $a_{ij} = 1$ if there is an edge connecting nodes v_i and v_j, and 0 otherwise. We consider simple graphs, i.e., undirected graphs without loops (i.e. $(v_i, v_j) = (v_j, v_i)$ and $(v_i, v_i) \notin E$) and without multiple edges (i.e. there is only one edge, if any, from vertex v_i to vertex v_j). Therefore A is a symmetric real matrix with zeros in its diagonal. The degree d_i of a node i is the number of its adjacent edges, i.e., $d_i = \sum_{j=1}^{n} a_{ij}$. The (unnormalized) Laplacian matrix of the graph is defined as $L = D - A$ where D is the diagonal matrix $D = \operatorname{diag}(d_1, d_2, \ldots, d_n)$. Some other definitions of the *Laplacian* matrix can be found in the literature; a good review of definitions can be found in [2]. A subgraph G' of G is a graph such that $G' = (E', V')$ with $V' \subseteq V$ and $E' \subseteq E$. A path is a sequence of nodes with an edge connecting every two consecutive nodes. A connected graph is a graph with a path connecting any pair of nodes; otherwise, the graph is said to be disconnected. Let $C_i \subseteq V$ be a set of nodes of a graph. We call $\pi_k = \{C_1, C_2, \ldots, C_k\}$ a partition of $G(V, E)$ when $V = \bigcup_{i=1}^{k} C_i$, and $\bigcap_{i=1}^{k} C_i = \emptyset$. A *connected component* of a graph $G = (V, E)$ is a connected subgraph $G_i(C_i, E_i)$ such that no other node in V can be added to C_i while preserving the property of being connected; i.e., a connected component is a maximal connected subgraph. We are interested in partitions π_k of a disconected graph $G(V, E)$ such that each subgraph $G_i(C_i, E_i)$ is a connected component.

We recall that a permutation matrix P is just the identity matrix with its rows reordered. Permutation matrices are orthogonal, i.e., $P^{\mathrm{T}} = P^{-1}$. A matrix $A \in \mathbb{R}^{n\times n}$, with $n \geqslant 2$, is said to be reducible[1] if there is a permutation matrix P of order n and an integer r with $1 \leqslant r \leqslant n-1$ such that $P^{\mathrm{T}}AP = \begin{bmatrix} B & C \\ 0 & H \end{bmatrix}$ where $B \in \mathbb{R}^{r\times r}$, $C \in \mathbb{R}^{r\times(n-r)}$, $H \in \mathbb{R}^{(n-r)\times(n-r)}$, and $0 \in \mathbb{R}^{(n-r)\times r}$ is the zero matrix. A matrix is said to be irreducible if it is not reducible. It is known (see, e.g., [14]) that the adjacency matrix A of a directed graph is irreducible if and only if the associated graph G is strongly connected. For an undirected graph, irreducibility implies connectivity. Note that the Laplacian $L = D - A$ is irreducible if and only if A is irreducible. In the following we recall some properties of L.

The Laplacian matrix is positive semidefinite if $x^{\mathrm{T}}Lx \geqslant 0$ for all $x \in \mathbb{R}^{n\times 1}$, or equivalently, L is symmetric and has all eigenvalues nonnegative. Since $L\mathbf{e}_n = \mathbf{0}$, with $\mathbf{e}_n$ the vector of all ones, L has 0 as an eigenvalue, and therefore it is a singular

[1] A matrix $A \in \mathbb{R}^{1\times 1}$ is said to be reducible if $A = 0$.

matrix. Since L is real symmetric, it is orthogonally diagonalizable, and therefore the rank of L is the number of nonzero eigenvalues of L. The eigenvalues of L can be ordered as

$$0 = \lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_{n-1} \leqslant \lambda_n. \tag{2.1}$$

It is known that $\lambda_2 = 0$ if and only if G is disconnected. In fact, λ_2 is called the *algebraic connectivity* of G. It is known (see, e.g., [19], [5], [20], [22]) that the number of connected components of a graph is given by the algebraic multiplicity of zero as an eigenvalue of the Laplacian matrix. Note therefore that the spectrum of L characterizes the connectivity properties of the associated graph G.

Note that L is irreducible if and only if G is connected. Therefore L is irreducible if and only if $\lambda_2 \neq 0$. Therefore, since the eigenvalues of L are ordered as in (2.1), L is irreducible if and only if $\operatorname{rank}(L) = n - 1$. It is known (see, e.g., [3]) that L is a singular M-matrix, i.e., L can be written as $L = sI - B$ with $B \geqslant 0$ and s the spectral radius of B.

We recall the following theorem due to Geiringer (see [30]) that we will use later.

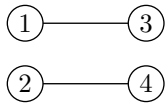
Theorem 2.1. *Let $A = [a_{ij}]$ be an $n \times n$ complex matrix, $n \geqslant 2$ and let $\mathcal{N} = \{1, 2, \ldots, n\}$. Then A is irreducible if and only if, for every two disjoint nonempty subsets S and T of $\mathcal{N}$ with $S \cup T = \mathcal{N}$, there exists an element $a_{ij} \neq 0$ of A with $i \in S$, $j \in T$.*

Note that for $n = 1$ we have the trivial case $A = 0 \in \mathbb{R}$ and $L = 0 \in \mathbb{R}$, which is a reducible matrix. The Laplacian matrix has been used extensively in *spectral clustering* (i.e, the technique of dividing a graph in connected components based on the eigenvectors) see [9], [28]. It is known that the inspection of the eigenvectors associated with λ_2 can lead to a partition of G in connected components. The seminal papers in this field are by Fiedler, see [7], [8]. The algorithm of the spectral bisection is given in [24]. A good explanation can be found in [23]. This technique can be also used to reduce the envelope-size (the sum of the row-widths), see [6], [17].

Example 2.1. Consider the toy graph in Figure 1. A simple computation shows that the spectrum of L is $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 2, \lambda_4 = 2$. The algebraic multiplicity of $\lambda = 0$ is two and therefore the graph has two connected components. An easy computation shows that the eigenspace of $\lambda = 0$ is spanned by the vectors

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

Note that the indices of the nonzero components of $\mathbf{v}_1$ and $\mathbf{v}_2$ give the labels of the nodes corresponding to each component of G.



$$L = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

Figure 1. Toy graph with two connected components and its corresponding Laplacian matrix.

3. The reverse Cuthill-McKee (RCM) algorithm

The Cuthill-McKee algorithm (CM) was introduced in [4] where the authors presented a scheme for numbering the nodes of a network in such a way that the corresponding adjacency matrix has a narrow bandwidth. They restricted their research to symmetric positive definite matrices and at that time they were motivated by systems of equations that resulted in this special kind of matrices, see [11] for details. To our knowledge the only limitation of their scheme is the symmetry of the adjacency matrix. The main motivation to obtain a narrow bandwidth was to reduce computer storage issues and calculation time. They explained that their objective was to determine a permutation matrix P such that the nonzero elements of PAP^{T} cluster about the main diagonal. Therefore, they focused on bandwidth minimization. Their strategy was to search for only few permutations P and then to choose the better one. In most cases (e.g., when the matrix can be transformed to band diagonal with no zero elements in the band) their scheme obtained the optimum numbering. The numbering obtained by their scheme corresponds, in graph theory, to the generation of a spanning tree (i.e., a subgraph of G which is a tree containing all the nodes of G, when G is connected). The algorithm selects a starting node and then visits all the neighbors in a level-by-level fashion, as in BFS. They remarked that as a result of their scheme one can easily check whether a matrix is reducible or not (we review this property in Section 5). In a later study, [10], the Reverse Cuthill-McKee algorithm (RCM) was introduced. RCM consists in the CM numbering but in reverse form. This simple procedure worked better than the original CM algorithm in some experimental studies while the bandwidth remained the same. A theoretical comparison of CM and RCM algorithms was given in [18] where it is shown that the reverse ordering is always as good as the original one in terms of storage and number of operations. RCM determines a starting node and looks for all the neighbors of this starting node. In the second step, the algorithm looks for the neighbors of these neighbors, and so on. The RCM algorithm can be outlined as follows:

Algorithm 1 RCM (Reverse Cuthill-McKee) for a network of n nodes.

1: select a starting node (e.g. a node with the lowest degree). Call it u_1
2: $k = 0$ (neighbors counter)
3: **for all** $i = 1$ to n **do**
4: **for** the h neighbors of u_i **do**
5: sort them in increasing order of degree
6: call them: u_{i+k+1}, u_{i+k+2}, ..., u_{i+k+h}
7: **end for**
8: $k = h$
9: **end for**
10: reverse the order of the n nodes

Note that when the order given by the RCM method is achieved, the construction of the matrix $P = (p_{ij})$ is straightforward: in row i, $p_{ij} = 1$ with i the new node label and j denoting the original one, see [4].

Since the introduction of the RCM method some new methods of reducing the bandwidth of a symmetric matrix have been introduced. We remark here that this problem is, in general, NP-complete [16]. These new methods can achieve better results in terms of bandwidth reduction but may be more expensive, in computational cost, than the RCM method (see, e.g., [17]). See, for example, [17] for methods like the RCM and methods that use hybrid techniques (spectral plus ordering) to reduce bandwidth. A review of some methods, including a spectral method that uses RCM as a preprocessor, can be found in [6]. For a comparison of reorderings, including RCM, in experiments with nonsymmetric matrices associated with the numerical solution of partial differential equations, see [1]. For a general view of reordering methods in linear systems, see [27]. A method to extend RCM to unsymmetric matrices is shown in [26]. RCM has been also used as an inspection tool for graph visualization [21].

In most mathematical packages there exists a function that provides the reordering produced by the RCM algorithm. For example, in MATLAB there exists the function *symrcm*. The expression $v = symrcm(A)$ computes an RCM ordering of A. This vector v is a permutation such that, using the syntax of MATLAB, $A(v, v)$ tends to have its nonzero entries closer to the diagonal. In MATLAB the algorithm first finds a certain vertex (a *pseudoperipheral* vertex) of the graph associated with the matrix A. Therefore the algorithm generates a level structure by BFS and orders the vertices by decreasing the distance from this initial vertex, see [12]. MATLAB claims that the implementation is based closely on the SPARSPAK implementation described in [11].

4. Laplacian matrix reordered by RCM

We are interested in applying the RCM algorithm to the Laplacian matrix L. To that end, we denote $\hat{L} = PLP^{\mathrm{T}}$ where P is the permutation matrix obtained by the RCM algorithm applied to L. We say that $\hat{L}$ is a Laplacian matrix of the graph reordered by RCM. It is known (see, e.g., [26], [27]) that $\hat{L}$ is block tridiagonal, i.e.:

$$\hat{L} = PLP^{\mathrm{T}} = \begin{bmatrix} \hat{L}_{11} & \hat{L}_{12} & \dots & 0 & 0 \\ \hat{L}_{21} & \hat{L}_{22} & \hat{L}_{23} & 0 & 0 \\ 0 & \hat{L}_{32} & \hat{L}_{33} & \hat{L}_{34} & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \hat{L}_{r,r-1} & \hat{L}_{r,r} \end{bmatrix}. \tag{4.1}$$

Example 4.1. Given

$$L = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}.$$

The RCM algorithm, computed by MATLAB, gives a permutation vector $\mathbf{v}^{\mathrm{T}} = [2, 3, 4, 1]$. That means, for example, that the old node 2 is now the new node 1, that is $p_{12} = 1$. Therefore the permutation matrix is

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

and $\hat{L}$ becomes the tridiagonal matrix

$$\hat{L} = PLP^{\mathrm{T}} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}.$$

We use the following notation: $\hat{l}_{ij}$ denotes the ij-element of $\hat{L}$, $\mathbf{1}_i \in \mathbb{R}^{i\times 1}$ is the column vector of all ones, $\mathbf{e}_i \in \mathbb{R}^{n\times 1}$ is the vector with ones in the first i entries and zero elsewhere. That is, $\mathbf{e}_i^{\mathrm{T}} = [\mathbf{1}_i^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}]$. For example, taking $n = 4$, $\mathbf{e}_2^{\mathrm{T}} = [1, 1, 0, 0]$.

We recall that the permutation matrices are orthogonal matrices: $P^{\mathrm{T}} = P^{-1}$. Since $\hat{L} = PLP^{\mathrm{T}}$, we say that L and $\hat{L}$ are similar matrices. Let $\mathbf{u}$ be an eigenvector

of $\hat{L}$ associated to the eigenvalue λ. That is, $\hat{L}\mathbf{u} = \lambda\mathbf{u}$. Therefore, it holds that $PLP^{\mathrm{T}}\mathbf{u} = \lambda\mathbf{u}$, that is to say $LP^{\mathrm{T}}\mathbf{u} = \lambda P^{\mathrm{T}}\mathbf{u}$ and thus λ is an eigenvalue of L with eigenvector $P^{\mathrm{T}}\mathbf{u}$. Therefore we conclude that L and $\hat{L}$ have the same spectrum and thus, the same rank. Therefore we have that L is irreducible if and only if $\hat{L}$ is irreducible. It is well known (see, e.g., [9]) that for a graph G of k connected components the corresponding Laplacian matrix L can be written as a block diagonal matrix. This fact can be derived from Theorem 2.1. What we want to remark in the following result is that the RCM method gives a permutation matrix P such that $\hat{L}$ is block diagonal. We give the proof for completeness and also to set the notation that we use later.

Lemma 4.1. *L is reducible if and only if $\hat{L}$ is a block diagonal matrix of the form*

$$\hat{L} = \begin{bmatrix} \hat{L}_{11} & & & \\ & \hat{L}_{22} & & \\ & & \ddots & \\ & & & \hat{L}_{kk} \end{bmatrix} \tag{4.2}$$

where each $\hat{L}_{ii}$, $i = 1, 2, \ldots, k$, is an irreducible matrix or a 1×1 zero matrix.

P r o o f. If L is reducible we know that G has $k \geqslant 2$ connected components with k the multiplicity of $\lambda = 0$. We know that the RCM method detects these k connected components. In fact, the RCM method packs the components. These connected components appear as irreducible blocks $\hat{L}_{ii}$, $i = 1, 2, \ldots k$ in $\hat{L}$; from Theorem 2.1 it is clear that $\hat{L}$ must be block diagonal. In the case of isolated nodes we have 1×1 null blocks in the diagonal. To prove the theorem in the other direction, note that it is clear that there exists a permutation matrix P, given by the RCM method, such that $\hat{L} = PLP^{\mathrm{T}}$ is block diagonal and therefore, by definition, L is reducible. □

In the following theorem we give a first characterization of the irreducibility of $\hat{L}$.

Theorem 4.1. *Let $n > 1$. $\hat{L}$ is irreducible if and only if $\hat{L}\,\mathbf{e}_i \neq \mathbf{0}$ for $i = 1, 2, \ldots, (n-1)$.*

P r o o f. Let $\hat{L}$ be irreducible. Let us assume that there exists $\mathbf{e}_i$, with $i < n$, such that $\hat{L}\mathbf{e}_i = \mathbf{0}$. Therefore $\mathbf{e}_i$ is an eigenvector for $\lambda = 0$. But since $\mathbf{e}_n$ is also an eigenvector for $\lambda = 0$ we conclude that the eigenspace for $\lambda = 0$ has a dimension greater than 1 and therefore, we conclude that the algebraic multiplicity of $\lambda = 0$ is greater than 1 and then the number of connected components in $\hat{L}$ is greater than 1 and then $\hat{L}$ is reducible, which is a contradiction. To prove the theorem in the opposite direction, let us assume that $\hat{L}$ is reducible. Therefore, L is also

reducible and from Lemma 4.1 we have that $\hat{L}$ must be block diagonal. Therefore there exists $i < n$ (with i the size of the first block, $\hat{L}_{ii}$) such that $\hat{L}\mathbf{e}_i = \mathbf{0}$, which is a contradiction. Therefore $\hat{L}$ is irreducible. □

We now present a new result from which a first method to detect components can be derived. We have seen in Lemma 4.1 that when L is reducible then $\hat{L}$ is block diagonal, with irreducible blocks $\hat{L}_{ii}$. In practice (for example, using MATLAB) the RCM method gives us a permutation vector to construct $\hat{L}$. The following result can be used to detect the sizes of the blocks $\hat{L}_{ii}$ when $\hat{L}$ is known.

Theorem 4.2. *Let $n > 1$. Let $p \in \mathcal{N}$. If there exists $p < n$ such that:*

$$\hat{L}\mathbf{e}_i \begin{cases} \neq \mathbf{0} & \text{for } i = 1, 2, \ldots, p-1, \\ = \mathbf{0} & \text{for } i = p \end{cases} \tag{4.3}$$

then $\hat{L}$ is a block diagonal matrix of the form

$$\hat{L} = \begin{bmatrix} \hat{L}_{11} & \\ & \tilde{L}_{22} \end{bmatrix} \tag{4.4}$$

with $\hat{L}_{11}$ a matrix of size $p \times p$ such that $\hat{L}_{11}$ is irreducible if $p > 1$ and a zero matrix if $p = 1$.

P r o o f. From Theorem 4.1 we have that $\hat{L}$ is reducible. Therefore L is reducible and from Lemma 4.1 we have that $\hat{L}$ is a block diagonal matrix of the form (4.2). We also know that $\hat{L}_{11}$ is an irreducible matrix. What we have to prove here is that the size of $\hat{L}_{11}$ is $p \times p$.

Let us assume that the size of $\hat{L}_{11}$ is $q \times q$, with $q > 1$, and study two cases:

Case 1. Suppose that $q < p$. Since $\hat{L}_{11}$ is itself a Laplacian matrix we have that $\mathbf{1}_q$ is an eigenvector associated with $\lambda = 0$, and therefore $\hat{L}_{11}\mathbf{1}_q = \mathbf{0}$ and then $\hat{L}\mathbf{e}_q = \mathbf{0}$ and this is a contradiction with the hypothesis (4.3).

Case 2. Suppose that $q > p$. The hypothesis we have is $\hat{L}\mathbf{e}_p = \mathbf{0}$, that is

$$\hat{L}\mathbf{e}_p = \begin{bmatrix} \hat{L}_{11} & 0 \\ 0 & \tilde{L}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{1}_p \\ \mathbf{0}_{n-p} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_p \\ \mathbf{0}_{n-p} \end{bmatrix}. \tag{4.5}$$

Let us make a partition of $\hat{L}_{11}$ in the form $\hat{L}_{11} = \begin{bmatrix} M & T \\ T^{\mathrm{T}} & N \end{bmatrix}$, with M of size $p \times p$. Then, from (4.5) we have

$$\hat{L}\mathbf{e}_p = \begin{bmatrix} M & T & 0 \\ T^{\mathrm{T}} & N & 0 \\ 0 & 0 & \tilde{L}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{1}_p \\ \mathbf{0}_{q-p} \\ \mathbf{0}_{n-q} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_p \\ \mathbf{0}_{q-p} \\ \mathbf{0}_{n-q} \end{bmatrix}. \tag{4.6}$$

Note that $T^{\mathrm{T}}\mathbf{1}_p = \mathbf{0}_{q-p}$ and since T has only nonpositive elements this implies $T = 0$ and therefore $\hat{L}_{11} = \begin{bmatrix} M & 0 \\ 0 & N \end{bmatrix}$ and thus $\hat{L}_{11}$ is reducible, which is a contradiction. Therefore we must have $q = p$, and since $\hat{L}_{11}$ is itself a Laplacian matrix we have $\hat{L}\,\mathbf{e}_p = \mathbf{0}$ in agreement with (4.3). The case $q = 1$ is trivial to prove. Therefore the proof is done. □

Note that with a recursive application of Theorem 4.2 we can compute the block diagonal form of $\hat{L}$ and thus the disconnected components of the graph G. Note that we have to apply the RCM method only once, since the submatrices $\hat{L}_{22}$, $\hat{L}_{33}$, etc., are results of the RCM method and we can apply Theorem 4.2 to each of them.

Example 4.2. Given

$$L = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}.$$

By applying the RCM algorithm to L we obtain that there exists a permutation matrix P such that

$$\hat{L} = PLP^{\mathrm{T}} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

and this matrix verifies conditions (4.3) with $p = 2$. According to Theorem 4.2 we have that $\hat{L}$ is a block diagonal matrix with irreducible blocks.

An easy computation shows that the eigenspace of $\lambda = 0$ is spanned by the vectors

$$\mathbf{w}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{w}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

Here we see that Theorem 4.2 is inspired by the search of an eigenvector as $\mathbf{w}_1$. Note, however, that the use of Theorem 4.2 avoids the explicit computation of the eigenspace of $\lambda = 0$ to detect the components.

Theorem 4.2 can be used to detect connected components when $\hat{L}$ is known. Nevertheless, by analysing some properties of the RCM method one can obtain an improved technique. We show this in the next section.

5. Properties of the RCM ordering

Definition (root of a component)**.** Given a connected component ordered by RCM each node has a label or index. We call the *root of the component* the node with the maximum index (that is, the node denoted by the greatest number). We denote by r the index of the root. This root corresponds to the starting node of the RCM algorithm shown in Section 3. Note that, by reversing the numbering, the root has the maximum index.

It is clear that: (a) a connected component has only one root, (b) for each node i (different from the root) of the connected component we have $i < r$, (c) if the connected component has n nodes, and the ordering begins in 1, then $r = n$. Note also that from each node i there is a path to the root r (since the component is connected, and thus irreducible).

Remark 5.1. Given a connected component ordered by RCM, each node i other than the root r is adjacent to a node $i + k \leqslant r$ for some $k \in \{1, 2, \ldots\}$.

Example 5.1. Let us consider the graphs shown in Figure 2. Graph a) is a possible RCM ordering since each node is connected to a node with higher index, except node 3, which is the root. In graph b) we see that node 2 is not connected to a node with higher index, and 2 is not the root, therefore this is not an ordering given by RCM.

a) b)

Figure 2. a) is an RCM ordering, while b) is not.

As a consequence of Remark 5.1 we have the following

Remark 5.2. Given a connected component ordered by RCM, let r be the root. Let $\hat{L}$ be the Laplacian of this connected component. Let $s_i(\hat{L})$ be the sum of the i-th row of $\hat{L}$ up to and including the diagonal, that is

$$s_i(\hat{L}) = \sum_{j=1}^{i} \hat{l}_{ij}, \quad i \in \mathcal{N}. \tag{5.1}$$

Then $s_i(\hat{L}) = 0$ if and only if $i = r$.

P r o o f. We denote $\hat{L} = PLP^{\mathrm{T}}$, $\hat{A} = PAP^{\mathrm{T}}$ and $\hat{D} = PDP^{\mathrm{T}}$. Since $L = D - A$ we have $\hat{L} = \hat{D} - \hat{A}$ with $\hat{l}_{ii} = \hat{d}_{ii} = \sum_{j=1}^{n} \hat{a}_{ij}$. Therefore

$$s_i(\hat{L}) = \sum_{j=1}^{i} \hat{l}_{ij} = \sum_{j=1}^{i-1} \hat{l}_{ij} + \hat{l}_{ii} = \sum_{j=1}^{i-1} \hat{l}_{ij} + \sum_{j=1}^{n} \hat{a}_{ij} = -\sum_{j=1}^{i-1} \hat{a}_{ij} + \sum_{j=1}^{n} \hat{a}_{ij} = \sum_{j>i}^{n} \hat{a}_{ij}.$$

In the last equality we have used that $\hat{a}_{ii} = 0$ for all i. From Remark 5.1 we have that each node i of the component, except the root, verifies $\hat{a}_{i,i+k} = 1$ for some $k \in \{1, 2, \ldots\}$ and therefore for all these nodes we have $s_i(\hat{L}) \neq 0$. The only node that verifies the equality $s_i(\hat{L}) = 0$ is the root, since it has no links to nodes with higher indices. □

Let us now show the properties of the RCM in a graph with more than one connected component.

Remark 5.3. Let $\hat{L}$ be the Laplacian of a graph with k connected components ordered by RCM. Let r_i, $i = 1, 2, \ldots k$, be the root of the i-th connected component. Then:

1. $s_i(\hat{L}) = 0$ if and only if $i = r_i$.
2. The nodes of a component i do not have a path to nodes with indices higher than r_i.

Lemma 5.1. *Let $\hat{L} \in \mathbb{R}^{n \times n}$, $n > 1$, be the Laplacian of a graph ordered by RCM. Let $s_i(\hat{L})$ be given by* (5.1). *Then $\hat{L}$ is irreducible if and only if $s_i(\hat{L}) = 0$ occurs only for $i = n$.*

P r o o f. If $\hat{L}$ is irreducible then there is only one component and from Remark 5.2 the proof follows. In the opposite direction, since $s_n(\hat{L}) = 0$, from Remark 5.2 we have that n is the root and from the basic properties we know that any node can follow a path to reach the root. Therefore the graph associated with the matrix $\hat{L}$ is connected and therefore $\hat{L}$ is irreducible. □

Proposition 5.1. *Let $\hat{L} \in \mathbb{R}^{n \times n}$ be the Laplacian of a graph ordered by RCM. Let $s_i(\hat{L})$ be given by* (5.1). *Let $n \geqslant 2$. Let $p \in \mathcal{N}$. If there exists $p < n$ such that:*

$$s_i(\hat{L}) \begin{cases} \neq 0 & \text{for } i = 1, 2, \ldots, p-1, \\ = 0 & \text{for } i = p \end{cases} \tag{5.2}$$

then $\hat{L}$ is a block diagonal matrix of the form

$$\hat{L} = \begin{bmatrix} \hat{L}_{11} & \\ & \tilde{L}_{22} \end{bmatrix} \tag{5.3}$$

with $\hat{L}_{11}$ a matrix of size $p \times p$ such that $\hat{L}_{11}$ is irreducible if $p > 1$ and a zero matrix if $p = 1$.

P r o o f. Since $s_i(\hat{L}) = 0$ for $i \neq n$ we have from Lemma 5.1 that $\hat{L}$ is reducible. From Lemma 4.1 we have that $\hat{L}$ is block diagonal with more than one block. Since

$s_p(\hat{L}) = 0$ we have from Remark 5.3 that the node p is the root of a component. The nodes of this component must have indices lower than p. Therefore, the p nodes of this component are the nodes described by $\hat{L}_{11}$. In the particular case that $p = 1$, $\hat{L}_{11}$ has only one element, which is zero; this means that node 1 is an isolated node. It is a component with a single element which is the root. □

By a recursive application of Proposition 5.1 one can derive an easy method to detect components. We show the details in the next section.

Example 5.2. Given the matrices L and $\hat{L}$ from Example 4.2 we have that $s_i(\hat{L}) = 0$ for $i = 2$ and $i = 4$. This means that there are two roots and therefore $\hat{L}$ has two components. $\hat{L}$ is a block diagonal matrix with two blocks: one formed by the nodes $1, 2$ and the other by the nodes $3, 4$ (with the numbering given by RCM, which is used in $\hat{L}$).

6. MATLAB code to detect connected components

A 5-lines MATLAB code to detect connected components can be obtained by using Proposition 5.1. The method can be written in the following form:

Algorithm 2 L-RCM Algorithm.

1: $L = sparse(diag(sum(A)) - A)$;
2: $rcm = symrcm(L)$;
3: $Lp = L(rcm, rcm)$;
4: $s = sum(tril(Lp), 2)$;
5: $cut = find(s == 0)$;

In the first line we construct the Laplacian matrix from the adjacency matrix. In the experiments we have used sparse matrices. The number of nonzero entries of A is $nnz := 2m$, with m being the number of edges. Since A is symmetric it suffices only to handle m entries. In sparse matrices, m is much less than n^2. The operation $sum(A)$ performs sums in n columns with an average of $2m/n$ entries per column, which totals $2m$ operations, therefore we need $O(2m)$. Note that $sum(A)$ is a vector of n entries, while $diag(sum(A))$ is a diagonal matrix of order n. In order to compute $diag(sum(A)) - A$, note that the entries are of the form $d_{ij} - a_{ij}$. When $i = j$ we have n operations of the form $d_{ii} - 0$, and when $i \neq j$ we have $2m$ operations of the form $0 - a_{ij}$. Thus the construction of $diag(sum(A)) - A$ has a cost of order $O(n+2m)$. To sum up, the construction of L in the first line has a cost of $O(n+4m)$.

The second line of the algorithm computes the RCM vector, that is, the ordering given by the RCM algorithm as computed by MATLAB. We have noted, in Section 3,

that MATLAB follows a procedure given in [11]. Therefore, as we have noted in Introduction, we can assume a time complexity of $O(q_{\max}m)$, where $q_{\max}$ is the maximum degree of any node.

The third line computes the Laplacian matrix of the graph ordered by the RCM method. This matrix is what we have called $\hat{L}$ in previous sections. Obviously, it would have a very high cost to compute $\hat{L}$ as the explicit matrix product PLP^{T}, with P the permutation matrix derived from the RCM. According to Matlab 2012 the cost of reordering a matrix, as in line 3, is proportional to nnz of that matrix. Therefore let us denote by $O(\gamma\, nnz(\hat{L}))$ the cost of computing Lp, with γ some parameter that depends on the implementation in MATLAB. Since $nnz(\hat{L}) = nnz(A) + n = 2m + n$ we have that the cost of the third line of the algorithm is of the order $O(\gamma(2m+n))$.

In line 4 the algorithm computes the sums $s_i(\hat{L})$ defined in equation (5.1). To compute these sums in MATLAB we take the lower triangular part of $\hat{L}$ and sum each row. Here we have m entries of A to sum plus the diagonal of L that counts n entries. Therefore this operation has a cost of $O(m+n)$.

Finally, in line 5 we find the indices of the sums that verify $s_i(\hat{L}) = 0$. Let us assume that this operation costs $O(n)$. The indices of these sums, the entries of the vector *cut*, give the location of the roots of the components, according to Proposition 5.1, i.e., vector *cut* gives the indices where there exists a gap in the blocks.

Therefore the total cost of the algorithm is of the order $O([q_{\max} + 2\gamma + 5]m + (3+\gamma)n)$.

The outputs of the method are the permutation vector *rcm* that defines the new ordering and the *cut* vector that identifies the blocks over the order defined by *rcm*.

In the following example we show how the method works. In [25] we show some applications of this method to some real graphs.

Remark 6.1. Since Lp is a symmetric matrix, line 4 of Algorithm 2 may be replaced by $s = sum(triu(Lp))$, which evaluates the column sums of the upper triangular part of Lp.

Example 6.1. Let us consider the graph shown in Figure 3.

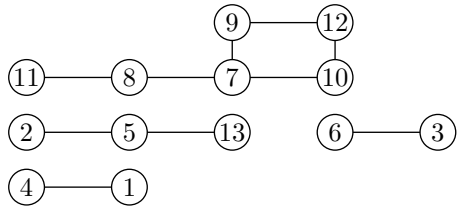


Figure 3. Graph with $n = 13$ and four connected components.

Matrix L is given by

$$L = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 2 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and the vector rcm is

$$rcm = [4, 1, 13, 5, 2, 6, 3, 11, 8, 7, 10, 9, 12].$$

Matrix $\hat{L}$ is given by

$$\hat{L} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 3 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

and the vector cut is $cut = [2, 5, 7, 13]^{\mathrm{T}}$. From the vectors rcm and cut we have the components (with the original labelling): $\{4, 1\}$, $\{13, 5, 2\}$, $\{6, 3\}$ and

$\{11, 8, 7, 10, 9, 12\}$. In Figure 4 we show the graph with the ordering given by the vector rcm.

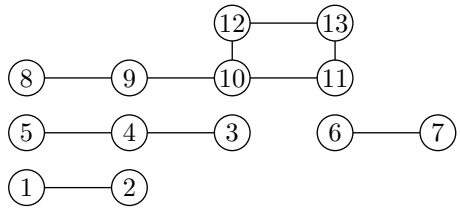


Figure 4. Graph with $n = 13$, ordered by RCM.

7. Conclusions

In this paper we have reviewed some works related to the Laplacian matrix that represents a graph and with the RCM algorithm. Our main contributions are some theoretical results concerning a Laplacian matrix that was reordered by the RCM algorithm. We illustrate the results with some examples. The reordering of the Laplacian matrix induced by the RCM algorithm reveals the number of the connected components in the graph. We have shown how some eigenvector properties of the Laplacian matrix inspire the use of certain row sums. One of the presented theoretical results serves as a basis for a 5-lines MATLAB code to detect connected components.

Acknowledgment. The authors would like to thank the anonymous referee for his/her valuable comments and remarks.

References

[1] *M. Benzi, D. B. Szyld, A. C. N. van Duin*: Orderings for incomplete factorization preconditioning of nonsymmetric problems. SIAM J. Sci. Comput. *20* (1999), 1652–1670.
[2] *D. Boley, G. Ranjan, Z.-L. Zhang*: Commute times for a directed graph using an asymmetric Laplacian. Linear Algebra Appl. *435* (2011), 224–242.
[3] *M. Bolten, S. Friedhoff, A. Frommer, M. Heming, K. Kahl*: Algebraic multigrid methods for Laplacians of graphs. Linear Algebra Appl. *434* (2011), 2225–2243.
[4] *E. Cuthill, J. McKee*: Reducing the bandwidth of sparse symmetric matrices. Proc. 24th Nat. Conf. of the ACM, ACM Publ P-69, Association for Computing Machinery, New York, 1969. pp. 157–172, doi:10.1145/800195.805928.
[5] *N. M. M. de Abreu*: Old and new results on algebraic connectivity of graphs. Linear Algebra Appl. *423,* (2007), 53–73.
[6] *G. M. Del Corso, F. Romani*: Heuristic spectral techniques for the reduction of bandwidth and work-bound of sparse matrices. Numer. Algorithms *28* (2001), 117–136.
[7] *M. Fiedler*: Algebraic connectivity of graphs. Czech. Math. J. *23* (1973), 298–305.
[8] *M. Fiedler*: A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. Czech. Math. J. *25* (1975), 619–633.

[9] *S. Fortunato*: Community detection in graphs. Phys. Rep. *486* (2010), 75–174.

[10] *J. A. George*: Computer Implementation of the Finite Element Method. Doctoral Dissertation, Stanford University, Stanford, 1971.

[11] *A. George, J. W.-H. Liu*: Computer Solution of Large Sparse Positive Definite Systems. Prentice-Hall Series in Computational Mathematics, Prentice-Hall, Englewood Cliffs, 1981.

[12] *J. R. Gilbert, C. Moler, R. Schreiber*: Sparse matrices in MATLAB: Design and implementation. SIAM J. Matrix Anal. Appl. *13* (1992), 333–356.

[13] *J. L. Gross, J. Yellen,* eds.: Handbook of Graph Theory. Discrete Mathematics and Its Applications, CRC Press, Boca Raton, 2004.

[14] *R. A. Horn, C. R. Johnson*: Matrix Analysis. Cambridge University Press, Cambridge, 1985.

[15] *D. Jungnickel*: Graphs, Networks and Algorithms. Algorithms and Computation in Mathematics 5, Springer, Berlin, 2008.

[16] *M. Juvan, B. Mohar*: Laplace eigenvalues and bandwidth-type invariants of graphs. J. Graph Theory, *17* (1993), 393–407.

[17] *G. Kumfert, A. Pothen*: Two improved algorithms for envelope and wavefront reduction. BIT *37* (1997), 559–590.

[18] *W-H. Liu, A. H. Sherman*: Comparative analysis of the Cuthill-McKee and the reverse Cuthill-McKee ordering algorithms for sparse matrices. SIAM J. Numer. Anal. *13* (1976), 198–213.

[19] *B. Mohar*: The Laplacian spectrum of graphs. Graph theory, Combinatorics, and Applications Vol. 2. Proc. Sixth Quadrennial International Conf. on the Theory and Applications of Graphs, Kalamazoo, Michigan, 1988 (Y. Alavi et all, eds.). John Wiley & Sons, New York, 1991, pp. 871–898.

[20] *J. J. Molitierno*: The spectral radius of submatrices of Laplacian matrices for graphs with cut vertices. Linear Algebra Appl. *428* (2008), 1987–1999.

[21] *C. Mueller, B. Martin, A. Lumsdaine*: A comparison of vertex ordering algorithms for large graph visualization. Visualization. Asia-Pacific Symposium on Visualization 2007, Sydney, Australia, 2007, pp. 141–148, doi: 10.1109/APVIS.2007.329289.

[22] *M. C. V. Nascimento, A. De Carvalho*: Spectral methods for graph clustering—a survay. Eur. J. Oper. Res. *211* (2011), 221–231.

[23] *M. E. J. Newman*: Networks. An Introduction. Oxford University Press, Oxford, 2010.

[24] *A. Pothen, H. D. Simon, K. P. Liou*: Partitioning sparse matrices with eigenvector of graphs. SIAM J. Matrix Anal. Appl. *11* (1990), 430–452.

[25] *M. Rebollo, C. Carrascosa, A. Palomares, F. Pedroche*: Some examples of detection of connected components in undirected graphs by using the Laplacian matrix and the RCM algorithm. Int. J. Complex Systems in Science *2* (2012), 11–15.

[26] *J. K. Reid, J.A. Scott*: Reducing the total bandwidth of a sparse unsymmetric matrix. Siam J. Matrix Anal. Appl. *28* (2006), 805–821.

[27] *Y. Saad*: Iterative Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, Philadelphia, 2003.

[28] *S. E. Schaeffer*: Graph clustering. Comput. Sci. Rev. *1* (2007), 27–64.

[29] *R. Tarjan*: Depth-first search and linear graph algorithms. SIAM J. Comput. *1* (1972), 146–160.

[30] *R. S. Varga*: Matrix Iterative Analysis. Springer Series in Computational Mathematics 27, Springer, Berlin, 2000.

Authors' addresses: Francisco Pedroche, Institut de Matemàtica Multidisciplinària, Universitat Politècnica de València, Cam'i de Vera s/n. 46022. València, Spain, e-mail: pedroche@imm.upv.es; Miguel Rebollo, Carlos Carrascosa, Alberto Palomares, Departament de Sistemes Informátics i Computació, Universitat Politècnica de València, Cam'i de Vera s/n. 46022. València, Spain, e-mail: mrebollo@dsic.upv.es, carrasco@dsic.upv.es, apalomares@dsic.upv.es.