

## Some examples of detection of connected components in undirected graphs by using the Laplacian matrix and the RCM algorithm

Miguel Rebollo<sup>1</sup>, Carlos Carrascosa<sup>1</sup>, Alberto Palomares<sup>1</sup> and Francisco Pedroche<sup>2,†</sup>

<sup>1</sup> *Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València*

<sup>2</sup> *Institut de Matemàtica Multidisciplinària, Universitat Politècnica de València*

**Abstract.** In this paper we use the recent method L-RCM, developed by the authors, to detect connected components in undirected graphs. The method uses the ordering RCM as a first step and the computation of the row sums of the Laplacian matrix as a second step. We make the computations in MATLAB. We show the use of the method in three examples.

*Keywords:* Connectivity, ordering algorithms, RCM, Graph Laplacian

*MSC 2000:* 68R10, 05C40, 05C78, 15B36

† **Corresponding author:** pedroche@imm.upv.es

**Received:** November 13th, 2012

**Published:** December 17th, 2012

### 1. Introduction

The authors have recently introduced a method to detect connected components in undirected graphs with no loops or multiple edges [3]. The method is called L-RCM since it uses as basic tools the Reverse Cuthill-McKee algorithm [2] and the Laplacian matrix. Given the adjacency matrix  $A$  of an undirected graph, the method L-RCM, in MATLAB notation, can be written as follows [3]:

1.  $L = \text{sparse}(\text{diag}(\text{sum}(A)) - A)$
2.  $rcm = \text{symrcm}(L)$
3.  $Lp = L(rcm, rcm)$

$$4. s = \text{sum}(\text{tril}(Lp), 2)$$

$$5. \text{cut} = \text{find}(s == 0)$$

where  $rcm$  is a vector with the labeling given by the RCM method. The number of entries of the vector  $cut$  gives the number of connected components. The entries of the vector  $cut$ , give the location of the *roots*<sup>1</sup> of the components as ordered by vector  $rcm$ .

The method has a theoretical computational cost of  $O(m + n)$ , where  $m$  is the number of edges and  $n$  the number of nodes. The examples that follow help understand the performance of the method.

## 2. Example 1. Network with 10 nodes and 4 components

Let us consider the network shown in figure 1 (left). The corresponding Laplacian matrix is

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

The  $rcm$  vector computed by MATLAB results

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

which means that the original node 6 is now node 1, the original node 9 is now node 2, etc. The network with the new labeling is shown in figure 1 (right).

The Laplacian matrix permuted with the order indicated in the  $rcm$  vector is

---

<sup>1</sup>We call *root* of a component the node with greatest index when ordered by RCM.

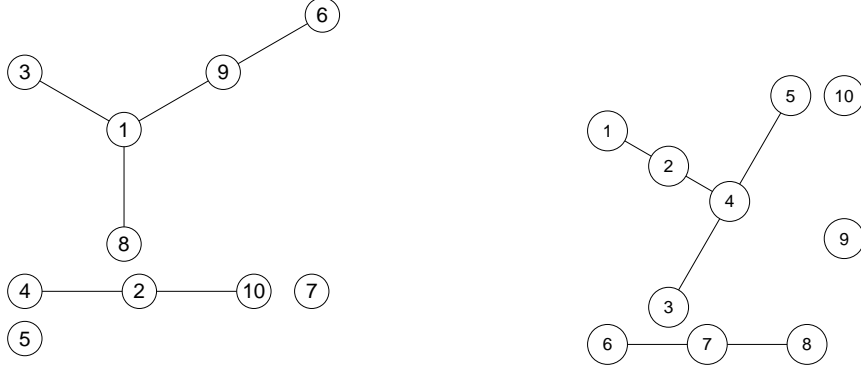


Figure 1: Network with 10 nodes and 4 components. Initial labeling and relabeling with RCM (right).

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

From  $Lp$  we see, for example, that in row number 5 the row sum up to the diagonal is zero. The same happens for the rows 8, 9, 10. Therefore we have  $cut = \{5, 8, 9, 10\}$ . Then the method L-RCM ensures that the network has four connected components. With the order given by RCM these components are:  $\{1, 2, 3, 4, 5\}$ ,  $\{6, 7, 8\}$ ,  $\{9\}$  and  $\{10\}$ . Note that the entries of vector  $cut$  locate the index in which a cut is produced.

### 3. Example 2. Network PGP

In this example we use the adjacency matrix corresponding to the network called PGP. This graph is the giant component of the network of users of the Pretty-Good-Privacy algorithm for secure information interchange, see

[1]. This network has  $n = 10680$  nodes, in 1 connected component, with 48632 nonzero entries. The number of edges is then  $m = 24316$ . Note that  $m + n = 34996$ . We use MATLAB version R2011b running in an HP-Z400 workstation. The method L-RCM takes 0.011 seconds to detect the component.

#### 4. Example 3. Days

In this example we use the network called *Days* (Reuters terror news network). It was obtained in Pajek format from networks produced by Steve Corman and Kevin Dooley at Arizona State University. It is Available at <http://vlado.fmf.uni-lj.si/pub/networks/data/CRA/terror.htm>. The network is based on news published by the news agency Reuters about the September 11 attack on the U.S. It has  $n = 13332$  nodes grouped in 22 connected components. The giant connected component is of size 13308. There are 3 components of size 2, and 18 isolated nodes. There are 296076 nonzero entries. The number of edges is then  $m = 148038$ . Note that  $m + n = 161370$ . In this case, the method L-RCM takes 0.048 seconds to detect the connected components.

#### 5. Conclusions

We have shown three examples of the application of the method L-RCM to detect connected components in undirected graphs. The method has a theoretical computational cost of  $O(m + n)$ . The computational costs in the examples shown agree with this theoretical estimate.

#### Acknowledgements

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

#### References

- [1] M. BOGUA, R. PASTOR-SATORRAS, A. DIAZ-GUILERA AND A. ARENAS. *Physical Review E*, **70**, 056122 (2004).
- [2] E. CUTHILL, AND J. MCKEE. *Proc. 24th Nat. Conf. of the ACM, ACM Publ P-69, Association for Computing Machinery, New York*, pp. 157–172 (1969).

- [3] F. PEDROCHE, M. REBOLLO, C. CARRASCOSA, AND A. PALOMARES  
<http://arxiv.org/abs/1206.5726> (submitted to *Discrete Applied Mathematics*), (2012).