Intensive Computation

8th april 2014

Exercise 1

Write the **function `epair`** that:

- takes as **input** a matrix
- computes the maximum eigenvalue and the corresponding eigenvector by implementing the Power method
- gives as **output** the maximum eigenvalue and the corresponding eigenvector

Exercise 2

Write a script that:

- writes a random adjacency matrix M of a graph representing a set of towns and the travel routes between these towns, having all diagonal elements equal to 1
- [optional] verify if the matrix M is ***primitive*** (A nonnegative square matrix $A=(a_{ij})$ is said to be *primitive* if there exists k such that $A^k \gg 0$, i.e., if there exists k such that for all i,j the entry (i,j) of A^k is positive)
- calls the **function `epair`** to obtain the principal eigenvector x1
- gives two subplots as **output** in a graphical window, one with the representation of the graph, the other with the graph where the most accessible town (or towns) is highlighted in green and the least accessible town (or towns) is highlighted in red; plot graphs using **`gplot.`**

Reference: http://matrixapps.blogspot.it/2010/07/gould-index-matrix-application-to.html

Exercise 3

Write a script that:

- writes a random adjacency matrix M of a graph
- computes the Laplacian matrix of M
- calls the **function `epair`** to obtain the principal eigenvector x1
- apply the Deflation method to obtain the Fiedler eigenvector (second eigenvector)
- determine the partitioned graph according to such an eigenvector
- gives two subplots as **output** in a graphical window, one with the representation of the graph, the other with the partitioned graph, using two different colors for nodes belonging to the two parts of the graph.