Intensive Computation – Homework 3

29th March 2019

Exercise 1

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

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

Write the **function `deflation`** that:

- takes as **input:** a matrix, the largest eigenvalue and the corresponding eigenvector
- computes the second largest eigenvalue and the corresponding eigenvector implementing one **Deflation method**
- gives as **output** the second eigenvalue and the corresponding eigenvector

Run a script calling the two functions **`epair`** and **`deflation`**, on a set of matrices and compare the obtained values with those produced by the function **`eig`** of Matlab.
Show the results in a table.

Exercise 2 – Gould index for eigenvector centrality and Fiedler eigenvector for Graph Partitioning

Write a script that:

- defines a **random adjacency matrix M** corresponding to a graph, representing a set of towns and the travel routes among these towns
- determines the most accessible town using the Gould index, calling the **function `epair`** to obtain the principal eigenvector x1 on the appropriately modified adjacency matrix
  Reference: http://matrixapps.blogspot.it/2010/07/gould-index-matrix-application-to.html)
- partitions the graph into two parts according to **the Fiedler eigenvector** (eigenvector corresponding to the second smallest eigenvector) obtained by considering the Laplacian matrix and using the **function `epair`** and the **function `deflation`**
- gives as **output** two subplots in a graphical window, one with the representation of the considered graph, the other with the graph where the most accessible town is highlighted and the two parts of the partitioned graph are represented using different colors.

Observations

- Use **`graph`** to define graphs and **`gplot`** to draw graphs.