**NUMERICAL ALGORITHMS, FALL 2017, HOMEWORK ASSIGNMENT 4**

Please hand in your work no later than 6 December 13:00h, in hardcopy (i.e., on paper). Make sure it is clear and legible, explain / motivate your answers, and put your name on it. For computer exercises, besides answers with motivation, please include a print-out of the code that you wrote for the problem (with comments/explanation), in Matlab or Python, as well as figures/tables to present your numerical results.

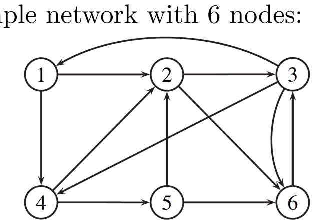
This homework assignment is an individual assignment, so you are expected to hand in original work. Work that is copied from someone else will not be accepted.

Please put your work in the mailbox of Raymond van Venetië (students in exercise class group A) or Fabian Stroh (students in exercise class group B) at the KdV Institute for Mathematics. Address: Science Park 105-107, 3rd floor (entrance via Nikhef), 1098 XG Amsterdam.

1. (a) Let $Q$ be a real, square orthogonal matrix. Show that all eigenvalues of $Q$ satisfy $|\lambda| = 1$.
   (b) A projection matrix is a square matrix $P$ for which $P^2 = P$. Find the eigenvalues of a projection matrix.

2. Generate a symmetric $4 \times 4$ matrix $\mathbf{A}$ randomly, e.g. by randomly generating the matrix $\mathbf{B}$ in Matlab using `B = randn(4,4)` and taking $\mathbf{A} = \mathbf{B} + \mathbf{B}^\mathbf{T}$. Use Rayleigh quotient iteration to compute the largest eigenvalue and corresponding eigenvector of $\mathbf{A}$. Use a suitable stopping criterion for your iterations (and explain why you think it is suitable).

3. Consider a very simple network with 6 nodes:




The link matrix $A$ for the network can be constructed by first creating a matrix $B$ with elements $b_{ij} = 1$ if node $j$ has a link to node $i$ and $b_{ij} = 0$ otherwise, followed by normalization so that the column sums of $A$ equal 1, i.e. $a_{ij} = b_{ij} / (\sum_i b_{ij})$. For our simple network the link matrix is

$$A = \begin{pmatrix} 0 & 0 & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 1 \\ \frac{1}{2} & 0 & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{3} & 0 & \frac{1}{2} & 0 \end{pmatrix}$$

   (a) Use power iterations to compute the largest eigenvalue $\lambda$ and corresponding eigenvector $u$ of $\mathbf{A}$. Start from a random vector $x$ with $x_i \geq 0$ and $\|x\|_1 = 1$. The *ranking* of the network nodes is determined by the ordering of $|u_i|$. Which node has the highest ranking (largest $|u_i|$)?
   (b) Suppose the network is slightly changed, so that node 3 no longer has links to nodes 1 and 4, only to 6. What is the link matrix of this changed network? What happens if you use power iterations to determine the eigenvector with the largest eigenvalue? Explain what is causing the problem.
   (c) *See next page*

(c) To tackle the problem encountered in (b), the link matrix can be modified by forming a convex combination of $A$ with a rank-one matrix. An example is $P = \alpha A + (1-\alpha) y e^T$ with $e$ a vector of all ones, $e = (1,1,1,..)^T$, and $y = (1/n)e$ (for our small network, $n = 6$). If $0 < \alpha < 1$, this modification adds a nonzero probability to go from a node to an arbitrary other node. Use power iterations to determine the eigenvector with the largest eigenvalue of $P$, for $\alpha = 0.95$ and for $\alpha = 0.75$. How do the eigenvector and the speed of convergence of the power iterations compare for these values of $\alpha$?

4. Let $A$ be a $n \times n$ tridiagonal matrix that results (up to an overall constant scaling factor) from an $n$-point spatial discretization of the Laplace operator in 1 spatial dimension. Specifically, all diagonal elements of $A$ have the value -2, all elements just below or above the diagonal have value 1.

   Thus, $A = \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & \dots \\ 1 & -2 & 1 & 0 & 0 & \dots \\ 0 & 1 & -2 & 1 & 0 & \dots \\ \vdots & & & & & \ddots \end{pmatrix}$.

   Take $n = 150$. Use QR iterations to compute the full eigenvalue spectrum of $A$. Compare against the eigenvalues obtained by the built-in Matlab routine `eig(A)`. Show in a figure the (approximate) eigenvalues after 10, 100 and 500 QR iterations, as well as the eigenvalues obtained with `eig(A)`.