Kungl. Tekniska Högskolan **2012**
CSC / NA
Elias Jarlebring

Deadline 2012-05-29

Homework set 3.
Advanced numerical methods for science and engineering - DN3250

http://www.csc.kth.se/~eliasj/DN3250/

QR-method and inverse iteration

---

1. In this exercise, we will compute some eigenvalues of the matrix

$$A = D \otimes I + I \otimes D \in \mathbb{R}^{n\times n},$$

where I denotes the $m \times m$ identity matrix and $\otimes$ denotes the Kronecker product, defined as

$$B \otimes C = \begin{pmatrix} b_{11}C & \dots & b_{1m}C \\ \vdots & \dots & \vdots \\ b_{m1}C & \dots & b_{mm}C \end{pmatrix}.$$

The matrix A can be constructed in matlab using

```
D=spdiags([e -2*e e], -1:1, m, m)/(m+1)^2;
A=kron(D,speye(m,m))+kron(speye(m,m),D);
```

(The matrix stems from the discretization of a two-dimensional Laplace operator on a square and can be solved analytically. The analytical solution will however not be needed for this exercise.)

(a) Carry out four iterations of the Rayleigh qoutient iteration for A with $m = 500$, i.e., $n = 250000$, started with $x_0 = (1, \dots, 1)$. Present the error as a function of iteration.

(b) A linear system with the structure

$$(D \otimes I + I \otimes D - \mu I)x = b$$

can be solved with the matlab command

```
x=reshape(lyap(D-0.5*mu*eye(m), -reshape(b,m,m)), m^2,1)
```

Use this to improve the Rayleigh qoutient iteration in part (a). Compare the CPU time with part (a).

2. (a) Implement Algorithm 28.1 in TB (QR-method without shifts) and apply it to the matrix generated by

```
n=100
A=expm((ones(n,n)+flipud(diag(1:n)))/n);
```

Visualize the error for the first 1000 iterations as a function of iteration by plotting the norm of the elements below the diagonal using `norm(tril(A,-1),1)`. Present also the total CPU time (using the commands `tic` and `toc`). In this exercise you may use the matlab command `qr`.

(b) Implement the following variation of the QR-method for the matrix in problem (a). First carry out the hessenberg reduction (Algorithm 26.1 in TB). After computing the hessenberg matrix, *impose* the expected structure, i.e., set those elements of the matrix which are expected to be zero (but are not zero due to rounding errors) to zero and change the representation to sparse. Apply the QR-method without shift (from part (a)) to the sparse matrix and visualize the result as in (a). Present also the total CPU time. Is this version better than part (a)? Justify your answer.