MAI0199 Computational Linear Algebra

Computer Exercise 2. Eigenvalues

1 General information

The assignment consists of a mixture of theoretical exercises and practical programming. At the end of the exercise a written report with well structured solutions to the theoretical questions and also Matlab programs, and graphs or plots that summarize the computational results should be sent by email to `Fredrik.Berntsson@liu.se`. In order to reduce the number of Matlab programs keep old code as comments when you modify a program in an exercise.

2 Hessenberg Reduction

The Hessenberg decomposition represents the closest to upper triangular shape that can be reached by a fixed sequence of Householder reflections. In the following exercises we hill use the matrix

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 2 & -1 & -4 \\ 3 & -1 & 3 & 1 \\ 4 & -4 & 1 & 4 \end{pmatrix}.$$

Exercise 2.1 Use the MATLAB routine `hess` to compute the Hessenberg decomposition $A = VHV^T$. Verify that A and H have the same eigenvalues. $\square$

3 The QR algorithm

In this section we will implement an algorithm for computing eigenvalues for a general matrix.

Exercise 3.1 The QR algorithm can be implemented in Matlab as follows:

```
B=A;
for k=1:N
    [Q,R]=qr(B);
    B=R*Q;
end;
```

Perform $N = 10$ steps in the QR algorithm. Compare `diag(B)` with the actual eigenvalues of the matrix A. Since A and B have the same eigenvalues we have found one eigenvalue of A with good accuracy. What is the error in the computed eigenvalue $B_{10}(4,4)$? $\square$

Exercise 3.2 Perform $N = 3$ iterations using the QR algorithm with shift $s = B_k(4,4)$. What is the error in the What is the error in the computed eigenvalue $B_3(4,4)$ now?

Type `eig(B(1:3,1:3)` and verify that the upper–left 3×3 block contains the remaining eigenvalues of B. Thus we can continue the QR iterations on a reduced size matrix. $\square$

Exercise 3.3 In this exercise we will compute all the eigenvalues of a matrix using the QR algorithm. Use the file `HessEigQR.m` as a starting point. Complete the code so that computes the QR–steps using Givens rotations.

Verify that the eigenvalues of the matrix A can be computed with good accuracy. How many QR steps are needed to compute each eigenvalue?

In order to speed up the convergence we will use shifts. Modify the code accordingly.How many QR steps are needed now? $\square$

Exercise 3.4 **Note:** The stopping criterion is $\|A(n, 1:n-1)\|_1 < \epsilon|A(n,n)|$. In this case $B(n,n)$ is considered an eigenvalue and the Gershgorin theorem provides a the error estimate $|A(n,n) - \lambda| < \epsilon|A(n,n)|$.

Is this a realistic stopping criteria? Can we expect all the computed eigenvalues to have a similar relative error? $\square$

Exercise 3.5 It may happen that some eigenvalues of A are complex. If so the QR algorithm may not converge. Propose a method to address this difficulty. $\square$

Exercise 3.6 Eigenvectors are not computed. Suppose λ is a single eigenvalue computed by the QR algorithm. Then $Ax = \lambda x$ has a non-trivial solution. Set $x_1 = 1$ and solve the resulting linear system to obtain the eigenvector x. Can this technique fail?

Explain how a similar technique can be used to obtain the eigenvectors corresponding to a double eigenvalue λ.

4 Roots of polynomials

In basic linear algebra courses eigenvalues are often computed by finding the roots of the *characteristic polynomial*, $p(\lambda) = \det(A - \lambda I)$. Finding the roots of a polynomial is a non-trivial task and the opposite problem is often more intressting. Namely, we have a polynomial $p(x)$ and want to find its roots.

Exercise 4.1 Show that the characteristic polynomial of the matrix

$$\begin{pmatrix} -c_{n-1} & -c_{n-2} & \dots & -c_1 & -c_0 \\ 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & 0 \end{pmatrix}.$$

is $p(x) = x^n + c_{n-1}x^{n-1} + \dots c_1 x + c_0$. $\square$

Exercise 4.2 Find the roots of the polynomial $p(x) = x^4 - 10x^3 + 35x^2 - 50x + 24$. Use either Matlabs `eig` or your own `HessEigQR` function. $\square$

Note: The matrix above is called the *companion* matrix and is often used in control theory to select eigenvalues, and thus the dynamical properties, of a linear system.