TANA15 Numerical Linear Algebra

Fredrik Berntsson (fredrik.berntsson@liu.se)

Computer Exercise 2. Computing Eigenvalues

1 General information

The programming assignments are intended to take more than two hours to complete. It is therefore important to be well prepared so that the computer sessions are used effectively. The assignment consists of a mixture of theoretical exercises and practical programming. A few of the exercises are based on custom Matlab programs or datasets that are provided in order to save time. These can be downloaded from the course website.

2 The power method

Let,

$$A = \begin{pmatrix} 2 & -1 & & & & \\ -1 & 3 & -1 & & & \\ & -1 & 4 & -1 & & \\ & & -1 & 5 & -1 & \\ & & & -1 & 6 & -1 \\ & & & & -1 & 7 \end{pmatrix}.$$

Exercise 2.1 Compute the eigenvalues of the matrix A using Matlabs `eig` function. Let the computed eigenvalues be $\lambda_1 > \lambda_2 \geq \ldots > \lambda_6$. Why are the eigenvalues real? The eigenvectors have a special property. Which one? □

Exercise 2.2 Compute an approximation of λ_1 using the *power method* and $N = 30$ iterations. Use the starting approximation $x_0 = (1, 1, \ldots, 1)^T$. Use the results obtained from the standard `eig` function as a comparison and compute the error.

Let ρ_k be the Rayleigh quotient at step k. Compute ρ_k, for $k = 3, 4$ and 5, and confirm that $|\rho_k - \lambda_1|/|\rho_{k-1} - \lambda_1| \approx \gamma^{k_1}$. What is γ and k_1 here?

Estimate the error in the approximation ρ_{30} using the formula $|\rho_{30} - \lambda_1| \leq \kappa_2(X)\|Ax_{30} - \rho_{30}x_{30}\|_2$. What is x and $\kappa_2(X)$ here? □

Exercise 2.3 Suppose we know that $\lambda_4 \approx 4$. Modify your program so that inverse iteration with shift $s = 4$ is used. Perform $N = 3$ iterations and compute an approximation ρ_3 of the eigenvalue λ_4. It is still true that $|\rho_k - \lambda_4|/|\rho_{k-1} - \lambda_4| \approx \gamma^{k_1}$. What is γ and k_1 now? $\square$

Note: The power method is rather slow and hence not widely used. However given a good shift s inverse iteration can converge very fast. Thus the method is often used for computing eigenvectors in the case when good approximations of the eigenvalues have been obtained using other methods.

3 Hessenberg factorization and the QR algorithm

In this section we will implement an algorithm for computing eigenvalues for a general matrix. As an example we will use the matrix

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

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

```
B1=B;
for k=1:N
    [Q,R]=qr(B1);
    B1=R*Q;
end;
```

Perform $N = 10$ steps in the QR algorithm. Compare `diag(B1)` with the actual eigenvalues of the matrix B. Since B_1 and B have the same eigenvalues we have found one eigenvalue of B 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(B1(1:3,1:3)` and verify that the upper–left $3{\times}3$ block contains the remaining eigenvalues of B. Thus we can continue the QR iterations on a reduced size matrix. $\square$

Computing the full QR–factorization is an expensive operation. Thus we want to find an orthogonal matrix Q such that $T = Q^T BQ$ has a structure that makes the QR factorization easier to compute.

A matrix T is said to have *Hessenberg* structure if $T_{i,j} = 0$ for indicies $i > j + 1$. A matrix B can be transformed into Hessenberg form using a series of Householder reflections. In the first step we apply a Householder reflection from the left to the block `B(2:N,1:N)`, and from the right to the block `B(1:N,2:N)`, to obtain

$$\tilde{B}_1 = P_1 B = \begin{pmatrix} x & x & x & x \\ + & + & + & + \\ & + & + & + \\ & + & + & + \end{pmatrix}, \quad \text{and,} \quad B_1 = (P_1 B)P_1^T = \begin{pmatrix} x & + & + & + \\ x & + & + & + \\ & + & + & + \\ & + & + & + \end{pmatrix}.$$

where x indicates the element changed during the operation and the o means that the element stayed the same. We can then procceed and apply a Householder reflection from the left to the block `B1(3:N,1:N)` and from the right to `B1(1:N,3:N)`, and so on, until we have obtained the desired *Hessenberg* structure.

Exercise 3.3 Copy the file `Hessenberg.m` from the course library. Complete the code so that the transformation to Hessenberg shape is computed. Type

```
>> H=Hessenberg(B);
>> eig(H)
```

Verify that H and B has the same eigenvalues. □

Exercise 3.4 Suppose T is an $N \times N$ Hessenberg matrix. Show how the QR factorization of T can be computed using only N Givens rotations. Furthermore demonstrate how one step of the QR algorithm can be implemented using $2N$ Givens rotations. □

Note: If the matrix B is symmetric then the corresponding Hessenberg matrix T is tridiagonal.

Exercise 3.5 On the course website there is a file `HessEigQR.m`. Complete the code so that computes the QR–steps. Verify that the eigenvalues of the matrix B can be computed with good accuracy. How many QR steps are needed to compute each eigenvalue?

Modify the code so that a shift is used. How many QR steps are needed now? □

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

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.

5 Ranking of Webpages

The popularity of Google is due to a page ranking algorithm that sorts search results better then its competitors do. The following is a simplified version of the Google page ranking algorithm. The pagerank x_i of page i is assumed to satisfy

$$x_i = \sum_{j \in P_i} \frac{x_j}{n_j},$$

where P_i is the set of pages linking to page i and n_j is the number of outgoing links from page j. On matrix form,

$$x = Ax,$$

and ranking pages is the same as computing the eigenvector x corresponding to the eigenvalue $\lambda = 1$. As an example, consider the web given in Figure 1. The pageranks of these pages satisfy

$$\begin{aligned} x_1 &= \tfrac{x_1}{3} + \tfrac{x_3}{3} + \tfrac{x_5}{2} \\ x_2 &= \tfrac{x_1}{3} + \tfrac{x_3}{3} \\ x_3 &= \tfrac{x_2}{1} + \tfrac{x_4}{2} + \tfrac{x_6}{3} \\ x_4 &= \tfrac{x_3}{3} + \tfrac{x_6}{3} \\ x_5 &= \tfrac{x_1}{3} + \tfrac{x_6}{3} \\ x_6 &= \tfrac{x_4}{2} + \tfrac{x_5}{2} \end{aligned} \quad \Leftrightarrow \quad x = Ax,\ A = \begin{pmatrix} 1/3 & 0 & 1/3 & 0 & 1/2 & 0 \\ 1/3 & 0 & 1/3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1/2 & 0 & 1/3 \\ 0 & 0 & 1/3 & 0 & 0 & 1/3 \\ 1/3 & 0 & 0 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}$$

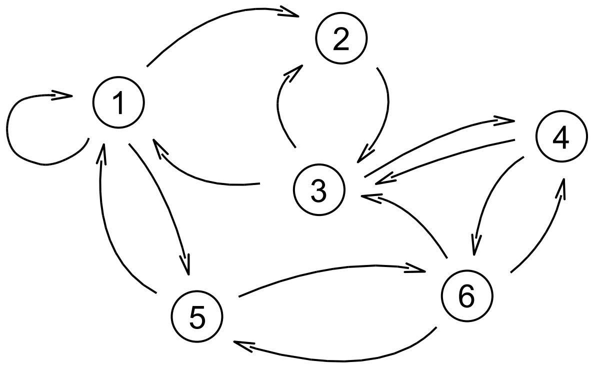

Figure 1: The link structure of a small collection of webpages.

Exercise 5.1 Compute the eigenvalues and eigenvectors of A. In what order does the page ranking algorithm rank the pages? $\square$

Exercise 5.2 It may happen that some eigenvalues of A are complex. If so the QR algorithm may not converge unless a suitable complex shift s_k is selected. Propose a method to address this difficulty. $\square$