8.3 Orthogonal Diagonalizability; Functions of a Matrix

Exercise 8.3. Let

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

(a) Find a matrix P that orthogonally diagonalizes the matrix A. You may use the MATLAB command *eig* and perform the Gram-Schmidt process. Use your result to find a diagonal matrix D satisfying $A = PDP^T$.

(b) Confirm that the matrix A satisfies its characteristic equation, in accordance with the Cayley-Hamilton theorem. You may use the symbolic object to find the characteristic polynomial and use the MATLAB command *coeffs* to find the coefficient of obtained characteristic polynomial.

(c) Find the spectral decomposition of A.

Solution.

(a)
```
A=[1/2 0 3/2 0; 0 1/2 0 3/2; 3/2 0 1/2 0; 0 3/2 0 1/2];

% V: eigen vector, D: eigen value
[V D]=eig(A);

% Gram-Schmidt process
P=GS_process(V);
disp('P is'); disp(P);
disp('D is'); disp(D);
disp('P_transpose is'); disp(P');
disp('P*D*P_transpose is'); disp(P*D*P');
disp('A is'); disp(A);
```

MATLAB results.

```
P is
   -0.7071         0         0   -0.7071
         0    0.7071    0.7071         0
    0.7071         0         0   -0.7071
         0   -0.7071    0.7071         0

D is
    -1     0     0     0
     0    -1     0     0
     0     0     2     0
```

```
     0     0     0     2

P_transpose is
   -0.7071         0    0.7071         0
         0    0.7071         0   -0.7071
         0    0.7071         0    0.7071
   -0.7071         0   -0.7071         0

P*D*P_transpose is
    0.5000         0    1.5000         0
         0    0.5000         0    1.5000
    1.5000         0    0.5000         0
         0    1.5000         0    0.5000

A is
    0.5000         0    1.5000         0
         0    0.5000         0    1.5000
    1.5000         0    0.5000         0
         0    1.5000         0    0.5000
```

(b)
```
% Symbolic variable lambda
syms lambda;

% Characteristic polynomial
char_poly=det(lambda*eye(size(A))-A);

% Expand the characteristic polynomial cf. simplify
polynomial=expand(char_poly);

% Coefficients extraction
coeff=coeffs(polynomial);

% According to the descending order of lambda degree
coefficient=coeff(end:-1:1);

 % Compute the matrix polynomial
poly_A=polyvalm(double(coefficient), A);

disp('Coefficients of the matrix characteristic polynomial is');
disp(double(coefficient));
disp('Matrix characteristic polynomial is'); disp(poly_A);
```

MATLAB results.

```
Coefficients of the matrix characteristic polynomial is
     1    -2    -3     4     4

Matrix characteristic polynomial is
```

```
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

(c)
```
[V D]=eig(A);
sum_A=0;
for i=1:size(A,1);
  % spectral decomposition
  sum_A=sum_A+D(i,i)*V(:,i)*V(:,i)';

  fprintf('lambda_%d is %f \n', i, D(i,i));
  fprintf('corresponding u_%d is \n', i);
  disp(V(:,i));
end

disp('spectral decomposition of A is'); disp(sum_A);
disp('A is'); disp(A);
```

MATLAB results.
```
lambda_1 is -1.000000
corresponding u_1 is
   -0.7071
         0
    0.7071
         0

lambda_2 is -1.000000
corresponding u_2 is
         0
    0.7071
         0
   -0.7071

lambda_3 is 2.000000
corresponding u_3 is
         0
    0.7071
         0
    0.7071

lambda_4 is 2.000000
corresponding u_4 is
   -0.7071
         0
   -0.7071
         0
```

```
spectral decomposition of A is
    0.5000         0    1.5000         0
         0    0.5000         0    1.5000
    1.5000         0    0.5000         0
         0    1.5000         0    0.5000

A is
    0.5000         0    1.5000         0
         0    0.5000         0    1.5000
    1.5000         0    0.5000         0
         0    1.5000         0    0.5000
```

8.4 Quadratic Forms

Exercise 8.4. (*Cholesky Factorization*)
In this problem, we find a Cholesky factorization of the Hilbert matrix

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

To generate the Hilbert matrix, you may use the MATLAB command *hilb*.

(a) Show that A is positive definite symmetric matrix by finding its eigenvalues and the MATLAB command *issymmetric*.

(b) Make a function file `ludecomp.m` to find the LU-decomposition of an invertible $n \times n$ matrix A such that A can be reduced to row echelon form by Gaussian elimination without row interchanges. You may refer to the four steps given in Page 157. Check your result by applying this function for the matrix given in the Example 2 of the Section 3.7.

(c) Referring to the Section 3.7, find the LDU-factorization of A from an LU-factorization of A by `ludecomp.m`.

(d) From the LDU-factorization of A obtained in (c), find a Cholesky factorization $A = R^T R$, where R is upper triangular.

(e) Use the MATLAB command *chol* to find a Cholesky factorization of A. Compare it with the result obtained in (d).

Solution.

(a)
```
format rat;
A=hilb(4);
```

```
eig_val=eig(A);
if all(eig_val>0) && issymmetric(A)==1
    disp('Given matrix A is'); disp(A);
    disp('A is symmetric and positive definite matrix.');
    fprintf('eigen value of A is '); disp(eig_val');
end
```

MATLAB results. Given matrix A is

```
       1              1/2            1/3            1/4
       1/2            1/3            1/4            1/5
       1/3            1/4            1/5            1/6
       1/4            1/5            1/6            1/7

A is symmetric and positive definite matrix.
eigen value of A is    66/682507   101/14989   262/1549   3500/2333
```

(b)

```
%--- This is a function file 'ludecomp.m' ---%
function [L, U] = ludecomp(A)

% The number of rows and columns of the matrix A.
[nrow, ncol] = size(A);

% Initialization of U and L.
U = A; L = eye(ncol);

% Forward Elimination %
for i=1:nrow
    % Find the first nonzero entry of the ith row.
    for k=i:ncol
        if U(i,k) ~= 0
            break % Terminates the execution of the loop.
        end
    end
    temp1 = U(i,k); % Save U(i,k) in temp.
    U(i,:) = (1/temp1) * U(i,:);
    % Normalize the pivot (i,k)-entry by 1 to the ith row.
    L(i,i) = (1/temp1)^(-1);
    % Place the reciprocal of the multiplier in that position in U.
        if i ~= nrow
        for j=(i+1):nrow
            temp2 = U(j,k); % Save U(j,k) in temp2.
            U(j,:) = ((-temp2) * U(i,:)) + U(j,:);
            % Add minus (j,k)-entry times the ith row to the jth row
            L(j,i) = -(-temp2);
            % Place the negative of the multiplier in that position in U.
        end
    end
```

```
end
```

(c)
```
% LU-factorization of A without row interchanges
[L, U] = ludecomp(A);

% From an LU-factorization of A, we can find the LDU-factorization of A,
% by appropriate normalization of L.
D = diag(diag(L));

for i = 1:4
    L(:, i) = L(:, i)./L(i, i);
end

disp('The LDU-factorization of A is');
disp('L = '); disp(L); disp('D = '); disp(D); disp('U = '); disp(U);
```

MATLAB results.

```
The LDU-factorization of A is
L =
       1              0              0              0
       1/2            1              0              0
       1/3            1              1              0
       1/4            9/10           3/2            1

D =
       1              0              0              0
       0              1/12           0              0
       0              0              1/180          0
       0              0              0              1/2800

U =
       1              1/2            1/3            1/4
       0              1              1              9/10
       0              0              1              3/2
       0              0              0              1
```

(d)
```
% From the LDU-factorization of A, find the Cholesky factor.
R1 = (L*sqrt(D))';
disp('The Cholesky factor R1 from the LDU-factorization of A is');
disp(R1);
```

MATLAB results.

```
The Cholesky factor R1 from the LDU-factorization of A is
       1              1/2            1/3            1/4
       0            390/1351       390/1351       351/1351
       0              0            317/4253       323/2889
       0              0              0            153/8096
```

(e)
```
% Find a Cholesky factorization of A by using the MATLAB command.
R2 = chol(A);
disp('The Cholesky factor R2 from the MATLAB command chol is');
disp(R2);
```

MATLAB results.

```
The Cholesky factor R2 from the MATLAB command chol is
       1              1/2             1/3             1/4
       0            390/1351        390/1351        351/1351
       0              0             317/4253        323/2889
       0              0               0             153/8096
```