Assignment 3

Due. In class Nov. 9 (Friday)

1. Write two Matlab functions for the LU decomposition of an m-by-n ($m \geq n$) matrix using Gaussian elimination without pivoting. Write the first function `LU2.m` in level 2 BLAS style and the second function `LU3.m` in level 3 BLAS style and compare their performances. The interface of `LU3.m` should look like:

```
function A = LU3(A,bsize)
% A = LU3(A,bsize)
%
% LU decomposition using Gaussian elimination without pivoting.
%
%       A - m-by-n (m>=n) matrix
%       bsize - block size
%       A is overwritten so that L*U = A, where
%         L = eye(m,n) + tril(A,-1) and U = triu(A(1:n,:))
%
% level-3 BLAS style
```

Solution

```
function A = LU2(A)
% A = LU2(A)
%
% LU decomposition using Gaussian elimination without pivoting.
%
%       A - m-by-n (m>=n) matrix
%       A is overwritten so that L*U = A, where
%         L = eye(n,m) + tril(A,-1) and U = triu(A(1:n,:))
%
% level-2 BLAS style

[m,n] = size(A);
for k=1:n-1
    A(k+1:m,k) = A(k+1:m,k)/A(k,k);       % multipliers
    % BLAS 2 rank-1 modification operation
    A(k+1:m,k+1:n) = A(k+1:m,k+1:n) - A(k+1:m,k)*A(k,k+1:n);
end
if m>n
    A(n+1:m,n) = A(n+1:m,n)/A(n,n);
end


function A = LU3(A,b)
% A = LU3(A,b)
```

```
%
% LU decomposition using Gaussian elimination without pivoting.
%
%       A - m-by-n (m>=n) matrix
%       b - block size
%       A is overwritten so that L*U = A, where
%         L = eye(m,n) + tril(A,-1) and U = triu(A(1:n,:))
%
% level-3 BLAS style
%
% dependency  LU2.m, level-2 BLAS LU decomposition

[m,n] = size(A);
for i=1:b:n-b
    endb = i + b - 1;   % end of the block
    A(i:m,i:endb) = LU2(A(i:m,i:endb));
    % solve lower triangular systems
%    A(i:endb,(i+b):n) ...
%  = (eye(b) + tril(A(i:endb,i:endb),-1)) \ A(i:endb,(i+b):n);
    for j=(i+b):n
        for k=(i+1):endb
            A(k:endb,j) = A(k:endb,j) - ...
                          A(k-1,j)*A(k:endb,k-1);
        end
    end
    % BLAS 3 matrix-matrix multiplication
    A((i+b):m,(i+b):n) = A((i+b):m,(i+b):n) - ...
                 A((i+b):m,i:endb)*A(i:endb,(i+b):n);
end
A((i+b):m,(i+b):n) = LU2(A((i+b):m,(i+b):n));
```

2. Let $\widehat{x}$ be the computed solution of the linear system $Ax = b$. In this exercise, we investigate the error

$$e = x - \widehat{x}$$

and the residual

$$r = b - A\widehat{x}.$$

Consider the system:

$$\begin{bmatrix} 0.563 & 0.217 \\ 0.658 & 0.255 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.780 \\ 0.913 \end{bmatrix}.$$

The exact solution of this system is $x_1 = 1$ and $x_2 = 1$. Solve the system using Gaussian elimination with partial pivoting on a hypothetical three-digit decimal computer in the nearest rounding mode and calculate the error e and the residual r. Explain your results.

Solution The pivot is 0.658 and multiplier is 0.856.

Updating 0.217, we get $0.217 - (0.856 \times 0.255) = -0.100 \times 10^{-2}$. Updating 0.780 in the right-hand side, we get $0.780 - (0.856 \times 0.913) = -0.200 \times 10^{-2}$. Solving for x:

$$x_2 = 0.200 \times 10 \quad x_1 = 0.612.$$

Obviously, the error e is large, however the residuals

$$r_1 = 0.100 \times 10^{-2} \quad r_2 = 0.000$$

are very small. Precisely because these two equations are almost linearly dependent, both residuals are small. The condition number is about 10^3.

3. In spring-mass mechanical vibration problem, we have a matrix of the form:

$$A = \begin{bmatrix} -a & -b \\ 1 & 0 \end{bmatrix}, \quad a, b \geq 0.$$

(a) Derive an algorithm for computing S and J in its Jordan canonical decomposition SJS^{-1}.

(b) Derive an algorithm for computing Q and T in its real Schur decomposition QTQ^{T}.
Solution (a) The characteristic polynomial is

$$\det \begin{bmatrix} \lambda + a & b \\ -1 & \lambda \end{bmatrix} = \lambda^2 + a\lambda + b,$$

which has the roots

$$\lambda_1 = \frac{-a - \sqrt{a^2 - 4b}}{2} \quad \text{and} \quad \lambda_2 = \frac{b}{\lambda_1}.$$

When $a^2 \neq 4b$, $\lambda_1 \neq \lambda_2$, solving

$$\begin{bmatrix} \lambda_i + a & b \\ -1 & \lambda_i \end{bmatrix} \mathbf{x}_i = 0,$$

the eigenvectors corresponding to λ_i are

$$\mathbf{x}_i = \begin{bmatrix} \lambda_i \\ 1 \end{bmatrix}.$$

Thus the Jordan form is

$$A = \begin{bmatrix} \lambda_1 & \lambda_2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & \lambda_2 \\ 1 & 1 \end{bmatrix}^{-1}.$$

When $a^2 = 4b$, $\lambda_1 = \lambda_2 = -a/2$. Solving

$$\begin{bmatrix} a/2 & b \\ -1 & -a/2 \end{bmatrix} \mathbf{x} = 0,$$

the eigenvector is

$$\mathbf{x} = \begin{bmatrix} \lambda \\ 1 \end{bmatrix},$$

which gives the first column of S. From

$$AS = S \begin{bmatrix} -a/2 & 1 \\ 0 & -a/2 \end{bmatrix},$$

the second column of S has the form:

$$\begin{bmatrix} 1+\gamma\lambda \\ \gamma \end{bmatrix}$$

Setting $\gamma = 0$, the Jordan form is

$$A = \begin{bmatrix} \lambda & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -a/2 & 1 \\ 0 & -a/2 \end{bmatrix} \begin{bmatrix} \lambda & 1 \\ 1 & 0 \end{bmatrix}^{-1}.$$

(b) When $a^2 < 4b$, A has two complex conjugate eigenvalues, then the real Schur form QTQ^{T} is given by $Q = I$ and $T = A$.

When $b = 0$, we have a quick return

$$Q = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad T = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix}.$$

Otherwise, from

$$AQ = Q \begin{bmatrix} t_{11} & t_{12} \\ 0 & t_{22} \end{bmatrix},$$

the first column of Q is a normalized eigenvector corresponding to the eigenvalue t_{11}. Setting

$$t_{11} = \lambda = \frac{-a - \sqrt{a^2 - 4b}}{2},$$

the corresponding eigenvector is $\mathbf{x} = [\lambda \;\; 1]^{\mathrm{T}}$. Construct an orthogonal

$$Q = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \quad \text{where} \quad \begin{bmatrix} c \\ -s \end{bmatrix} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}.$$

Then the upper triangular T in the Schur form is given by

$$t_{11} = \lambda, \quad t_{21} = 0, \quad t_{22} = \frac{b}{\lambda},$$

and it can be verified that

$$t_{12} = [c \;\; -s]\, A \begin{bmatrix} s \\ c \end{bmatrix} = -b - 1.$$

4. Using the identity

$$\sin\alpha - \sin\beta = 2\sin\frac{\alpha-\beta}{2}\cos\frac{\alpha+\beta}{2},$$

find the singular values σ_k and singular vectors $\mathbf{v}_k$ of the $(n+1)$-by-n matrix:

$$A = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & 0 & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & 0 & & -1 & 1 \\ & & & & -1 \end{bmatrix}.$$

Then find its singular vectors $\mathbf{u}_k$.
The matrix U is call the discrete consine transform (DCT) and V called the discrete sine transform (DST), comparing with the discrete Fourier transform (DFT). The SVD of A shows the relation between DST and DCT.
Form the n-by-n matrix $A^{\mathrm{T}}A$. Now you know its eigenvalues and eigenvectors.
Solution From

$$\begin{aligned} \sin\frac{k\pi}{n+1} &= 2\sin\frac{k\pi}{2(n+1)}\cos\frac{k\pi}{2(n+1)} \\ \sin\frac{(i+1)k\pi}{n+1} - \sin\frac{ik\pi}{n+1} &= 2\sin\frac{k\pi}{2(n+1)}\cos\frac{(2i+1)k\pi}{2(n+1)}, \quad i = 1,2,...,n-1 \\ -\sin\frac{nk\pi}{n+1} &= 2\sin\frac{k\pi}{2(n+1)}\cos\frac{(2n+1)k\pi}{2(n+1)} \end{aligned}$$

we have

$$A\tilde{\mathbf{v}}_k = \sigma_k\tilde{\mathbf{u}}_k,$$

where

$$\sigma_k = 2\sin\frac{k\pi}{2(n+1)}, \quad \tilde{\mathbf{v}}_k = \begin{bmatrix} \sin\frac{k\pi}{n+1} \\ \sin\frac{2k\pi}{n+1} \\ \vdots \\ \sin\frac{(n-1)k\pi}{n+1} \\ \sin\frac{nk\pi}{n+1} \end{bmatrix}, \quad \tilde{\mathbf{u}}_k = \begin{bmatrix} \cos\frac{k\pi}{2(n+1)} \\ \cos\frac{3k\pi}{2(n+1)} \\ \vdots \\ \cos\frac{(2n-1)k\pi}{2(n+1)} \\ \cos\frac{(2n+1)k\pi}{2(n+1)} \end{bmatrix}, \quad k = 1,2,...,n.$$

It can be verified that $\tilde{\mathbf{v}}_k$ are the eigenvectors of

$$A^{\mathrm{T}}A = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}$$

and the corresponding eigenvalues are σ_k^2. Consequently, $\tilde{\mathbf{v}}_k$ are orthogonal. Alternatively, we can prove that $(\sigma_k^2\tilde{\mathbf{v}}_k)^{\mathrm{T}}\tilde{\mathbf{v}}_j = \tilde{\mathbf{v}}_k^{\mathrm{T}}(\sigma_j^2\tilde{\mathbf{v}}_j)$, which implies that $\tilde{\mathbf{v}}_k^{\mathrm{T}}\tilde{\mathbf{v}}_j = 0$, when $k \neq j$. The normalized $\mathbf{v}_k = \tilde{\mathbf{v}}_k/\|\tilde{\mathbf{v}}_k\|_2$, where $\|\tilde{\mathbf{v}}_k\|_2 = \sqrt{(n+1)/2}$.