Lab Session 3

MA-423 : Matrix Computations Lab 2023 R. Alam

MATLAB Scripts and functions for $Ax = b$

1. Write a MATLAB function that implements GENP. Your function should look like the following.

```
function [L, U] = GENP(A);
% [L U] = GENP(A) produces a unit lower triangular matrix L and an upper
% triangular matrix U so that A= LU.

[n, n] = size(A);
for k = 1:n-1
    % compute multipliers for k-th step
    A(k+1:n,k) = A(k+1:n,k)/A(k,k);
    % update A(k+1:n,k+1:n)
    j = k+1:n;
    A(j,j) = A(j,j)-A(j,k)*A(k,j);
end
% strict lower triangle of A, plus I
L = eye(n,n)+ tril(A,-1);
U = triu(A); % upper triangle of A
```

Next, write a MATLAB function that implements GEPP. Your function should look like the following.

```
function [L, U, p] = GEPP(A);
% [L U, p] = GEPP(A) produces a unit lower triangular matrix L, an upper
% triangular matrix U and a permutation vector p, so that A(p,:)= LU.

[n, n] = size(A);
p = (1:n)';
for k = 1:n-1

    % find largest element in A(k:n,k)
    [r, m] = max( abs( A(k:n,k) ) );
    m = m+k-1;
    if (m ~=k) % swap row
        A([k m],:)  = A([m k],:);
        p([k m]) = p([m k]);
    end
        if (A(k,k) ~= 0)
```

```
            % compute multipliers for k-th step
            A(k+1:n,k) = A(k+1:n,k)/A(k,k);
            % update A(k+1:n,k+1:n)
            j = k+1:n;
            A(j,j) = A(j,j)-A(j,k)*A(k,j);
        end
    end
    % strict lower triangle of A, plus I
    L = eye(n,n)+ tril(A,-1);
    U = triu(A); % upper triangle of A
```

Now modify GEPP to write a new function GEPP2 that also computes determinant of A.

```
function [L, U, p, d] = GEPP2(A);
% [L, U, p, d] = GEPP(A) produces a unit lower triangular matrix L, an upper
% triangular matrix U and a permutation vector p, so that A(p,:)= LU
% and d = det(A).
```

Finally write a MATLAB function that implements GECP

```
function [L, U, p, q] = GECP(A);
% [L, U, p, q] = GECP(A) produces a unit lower triangular matrix L,
% an upper triangular matrix U and two permutation vectors p and q,
% so that A(p,q)= LU.
```

2. Consider $A = \begin{bmatrix} 10^{-16} & 1 \\ 1 & 1 \end{bmatrix}$. Compute `[L, U] = GENP(A)`and define `E = L*U-A`. Also compute `[L, U, p] = GEPP(A)` and compute `F = LU - A(p,:)`. Now compute the norms of `E` and `F` using the command `norm(E)` and `norm(F)`. Note that `norm(E)/norm(A)` and `norm(F)/norm(A)` are backward errors of GENP and GEPP. Are GENP and GEPP backward stable? Recall that if `[L, U]= ALG(A)` then `ALG` is backward stable if `A+E = LU` for some `E` such that `norm(E)/norm(A)` $= \mathcal{O}(\mathbf{u})$.

   Next, consider `b = A* ones(2,1)` and solve $Ax = b$ using GENP and GEPP. Let `xn` and `xp` be the computed results. Note that $x = [1, 1]^\top$ is the exact solution of `Ax= b`. Compute the relative errors `errn = norm(xn-ones(2,1))/norm(ones(2,1))` and `errp = norm(xp-ones(2,1))/norm(ones(2,1))`. Which method produces better result?

   An $n \times n$ Hilbert matrix H is given by $H(i, j) = 1/(i+j-1)$. The command `H = hilb(n)` generates H. Compute determinant of 8×8 Hilbert matrix H using GEPP2 as well as the command `det(H)`. Do you get the same result?

3. **Assignment:**If $\hat{x}$ is the computed solution of $Ax = b$ then $r := A\hat{x} - b$ is called the *residual*. Of course $r = 0$ if and only if $x = \hat{x}$. But usually $r \neq 0$. Does a small $\|r\|_2$ imply $\|x - \hat{x}\|_2$ small? Try the following experiment.

   Consider the Hilbert matrix H (the MATLAB command `H = hilb(n)` generates H) and perform the following computations:

```
>> n=10;
>> H=hilb(n); x = randn(n,1);
>> b = H*x;
```

```
>> x1= H \ b;
>> r = H*x1-b;
>> disp( [norm(r) norm(x-x1)])
```

Does small residual imply small error? What is your conclusion?

Compute LU factorization `[L, U]= GENP(H)` of the Hilbert matrix H for $n = 8, 10, 12$ and check the backward stability of GENP. Recall that if `[L, U]= GENP(H)` then `GENP` is backward stable if `H+E = LU` for some `E` such that `norm(E)/norm(H)` $= \mathcal{O}(\mathbf{u})$. What is your conclusion?

Now choose `x = randn(n,1)` and `b = H*x`. Solve `Hx = b` using `L` an `U` computed by GENP for $n = 8, 10, 12$. This can be done as `x = U\(L\b)`. Compute the relative errors `norm(x-ones(n,1))/norm(ones(n,1))` and draw your conclusion about accuracy of the computed solution. **10 marks**

*** End ***