Practical Worksheet 02 – Solving Triangular Systems

Exercise 01

1. Write a Matlab function "is_upper_triangular" that takes a matrix ***A*** as an argument and returns 1 if matrix ***A*** is an upper triangular matrix, and 0 otherwise (Matrix ***A*** is said to be upper triangular if and only if all elements below its diagonal are zero). A diagonal matrix is also considered an upper triangular matrix.

```
% Solution 01
function b = est_triangulaire_superieur(A)
    [m n] = size(A);
    if m == n
        b = 1;
        for i = 2:m
            for j = 1:i-1
                if A(i, j) ~= 0
                    b = 0;
                end
            end
        end
    else
        b = 0;
        display('La matrice n''est pas carree');
    end
end

% Solution 02
function b = est_triangulaire_superieur(A)
    [m n] = size(A);
    if m == n
        b = isequal(triu(A),A)
    else
        b = 0;
        display('La matrice n''est pas carree');
    end
end
```

2. Write a Matlab function "is_lower_triangular" that takes a matrix ***A*** as an argument and returns 1 if matrix ***A*** is a lower triangular matrix, and 0 otherwise (Matrix ***A*** is said to be lower triangular if and only if all elements above its diagonal are zero). A diagonal matrix is also considered a lower triangular matrix.

```
% Solution 01
function b = est_triangulaire_inferieur(A)
    [m n] = size(A);
    if m == n
        b = 1;
        for i = 1:m
            for j = i+1:n
                if A(i, j) ~= 0
                    b = 0;
                end
            end
        end
    else
        b = 0;
        display('La matrice n''est pas carree');
    end
end

% Solution 02
function b = est_triangulaire_inferieur(A)
    [m n] = size(A);
    if m == n
        b = isequal(tril(A), A)
    else
        b = 0;
        display('La matrice n''est pas carree');
    end
end
```

3. Write a Matlab function "matrix_type" that takes a matrix ***A*** as an argument and returns one of the following results:
   - 0 if matrix ***A*** is not triangular.
   - 1 if matrix ***A*** is upper triangular.
   - 2 if matrix ***A*** is lower triangular.
   - 3 if matrix ***A*** is a diagonal matrix.

   Test the function "matrix_type" on a non-triangular matrix, an upper triangular matrix, a lower triangular matrix, and a diagonal matrix of your choice and check if it returns the correct result in each situation.
   Hint: to check that a matrix is diagonal, you have to test if it is upper and lower triangular at the same time.

```
function t = type_matrice(M)
    if est_triangulaire_inferieur(M) &&
                est_triangulaire_superieur(M)
        t = 3;
    elseif est_triangulaire_inferieur(M)
        t = 2;
    elseif est_triangulaire_superieur(M)
        t = 1;
    else
        t = 0;
    end
end
```

Exercise 02

1. Write a Matlab function "determinant_triangulaire" that takes a matrix A as an argument and calculates and returns the determinant of A if it is triangular (or diagonal), otherwise the function should display the following message: "Matrix A is not triangular".
    - Test your function on a triangular matrix of your choice, and compare the result obtained by your function to that obtained by the predefined Matlab function "**det**" to check if it returns the right result.

```
% Solution 01
function d = determinant_triangulaire(A)
    [m n] = size(A);
    if m == n
        if type_matrice(A) > 0
            d = 1;
            for i = 1:m
                d = d * A(i, i);
            end
        else
            display('La matrice non triangulaire');
        end
    else
        display('La matrice n''est pas carree');
    end
end

% Solution 02
function d = determinant_triangulaire(A)
    [m n] = size(A);
    if m == n
        if type_matrice(A) > 0
            d = prod(diag(A));
```

```
            else
                display('La matrice non triangulaire');
            end
        else
            display('La matrice n''est pas carree');
        end
    end
```

2. Write an "invertible" Matlab function that takes a matrix A as an argument and returns 1 if A is invertible, and 0 otherwise.
   Hint: use the functions declared in exercise 01.

```
function inv = triangulaire_inversible(A)
    [m n] = size(A);
    if m == n
        if type_matrice(A) > 0
            inv = determinant_triangulaire(A) == 0;
        else
            inv = 0;
            display('La matrice non triangulaire');
        end
    else
        inv = 0;
        display('La matrice n''est pas carree');
    end
end
```

Exercise 03

1. Write a Matlab function “solve_upper_triangular” that takes as an argument an upper triangular matrix with coefficients A and a right-hand side vector b and that solves the upper triangular system Ax = b.

```
function x = resoudre_triangulaire_superieur(A, b)
    n = size(A, 1);
    x = zeros(n, 1);
    x(n) = b(n) / A(n, n);
    for i = n-1:-1:1
        % Opération vectorielle (commentée)
        % x(i) = (b(i) - x(i+1:n)*A(i,i+1:n)) / A(i, i);
        % Ou utiliser les boucles
        s = b(i);
        for j = i+1:n
            s = s - A(i, j) * x(j);
        end
```

```
            x(i) = s / A(i, i);
        end
    end
```

2. Write a Matlab function "solve_lower_triangular" that takes as an argument a lower triangular matrix with coefficients A and a right-hand side vector b and that solves the lower triangular system Ax = b.

```
function x = resoudre_triangulaire_inferieur(A, b)
    n = size(A, 1);
    x = zeros(n, 1);
    x(1) = b(1) / A(1, 1);
    for i = 2:n
        % Opération vectorielle (commentée)
        % x(i) = (b(i) - x(1:i-1)*A(i,1:i-1)) / A(i, i);
        % Ou utiliser les boucles
        s = b(i);
        for j = 1:i-1
            s = s - A(i, j) * x(j);
        end
        x(i) = s / A(i, i);
    end
end
```

3. Write a Matlab script that allows the user to:
    - Enter n the size of a linear system to be solved.
    - Enter the elements of the matrix A and the right-hand side b.
    - If the matrix A is triangular (or diagonal) and invertible, solve the triangular system Ax = b according to the nature of the matrix A (upper or lower triangular) and display the result. Otherwise, the script must display one of the following messages:
        i. "Matrix A is not triangular": if the matrix A is not triangular.
        ii. "Matrix A is not invertible": if matrix A is triangular but not invertible.
        iii. Test the script on a matrix A and a vector b of your choice, and compare the result to that obtained by the Matlab instruction "x = A\b" to verify that your code is correct.

    Hint: use the functions defined in exercises 01 and 02.

```
n = input('Taille du systeme lineaire a resoudre : ');
display('Matrice de coefficients A');
A = zeros(n, n);
for i = 1:n
    for j = 1:n
        A(i, j) = input(['A(', num2str(i), ', ', num2str(j),
') : ']);
    end
end
if type_matrice(A) == 0
    display('Matrice non triangulaire');
elseif inversible(A) == 0
    display('Matrice non inversible');
else
    display('Second membre b');
    b = zeros(n, 1);
    for i = 1:n
        b(i) = input(['b(', num2str(i), ') : ']);
    end
    if est_triangulaire_superieur(A)
        x = resoudre_triangulaire_superieure(A, b);
    else % ça marche même si la matrice est diagonale
        x = resoudre_triangulaire_inferieure(A, b);
    end
    display(x);
end
```