Homework Assignment 5 due by Thursday 11:45 PM, November 28, 2019. Test your answers on MATLAB Grader or Live Editor on your MATLAB.

1. Create a function `gaussian_elimination` that performs the gaussian elimination of linear system of the form $Ax = b$. The function should return the associated upper triangular matrix $U$and the modified right-hand side f. The function header should look something like

   ```
   function [U,f] = gaussian_elimination(A,b)
   ```

   Create a function `backward_substitution` that solve the system $Ux = f$ when U is an upper triangular matrix. The function header should look something like

   ```
   function x = backward_substitution(U,f)
   ```

   Let us recall that the output has to be a vector.

   Apply the two functions to solve the 7×7 linear system

$$\begin{bmatrix} 6 & 1 & 7 & 7 & 6 & 1 & 8 \\ 7 & 5 & 5 & 2 & 7 & 8 & 7 \\ 9 & 5 & 3 & 4 & 4 & 4 & 3 \\ 9 & 8 & 1 & 6 & -1 & 8 & 2 \\ 1 & 4 & 5 & 3 & 3 & 3 & 7 \\ 1 & 3 & 2 & 7 & 4 & 7 & 9 \\ 6 & 6 & -2 & 3 & 2 & 3 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \\ 8 \\ 7 \\ 1 \\ 8 \\ 9 \end{bmatrix} \tag{1}$$

   Call, `U,f,x` the results. compare your solution to `x0` obtained with the Matlab command `A\b`. Call `Err` the absolute error. Comment using `%`.

2. Create a function `LU_factorization` that performs the LU factorization of a matrix A. You may use the previous functions you implemented. The function header should at least contain

   ```
   function [L,U] = LU_factorization(A)
   ```

   Create a function `forward_substitution` that solve the system $Ly = b$, when L is a lower triangular matrix. The function header should look something like

   ```
   function y = forward_substitution(L,b)
   ```

   Let us recall that the output has to be a vector.
   Use those two functions and the function `backward_substitution` to solve (1). Do you find the same solution ? Call, `L,U,y,x` the results. compare your solution to `x0` obtained with the Matlab command `A\b`. Call `Err` the absolute error. Comment using `%`.

3. Write a function called `gauss_seidel` that inputs an $n \times n$ matrix, A, a column vector, $\mathbf{b}$, an initial guess $\mathbf{x}^{(\mathbf{0})}$, an error tolerance ϵ, and a maximum number of iterations, and output an approximate solution obtained using the Gauss-Seidel method, the error and the number of iterations. The header should look like `[x, err, N] = gauss_seidel(A, b, x0, tol, Nmax)`. Use the method to find approximate solutions to the linear system

$$\begin{bmatrix} -2 & 1 & 0 & 0 & 0 \\ 2 & 10 & -2 & 0 & 0 \\ 0 & 2 & -4 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & -3 & -7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} -5 \\ 3 \\ 3 \\ -7 \\ -8 \end{bmatrix}$$

to within an accuracy of $\epsilon = 10^{-5}$, with a maximum of $N = 100$ iterations. If your method succeeds, report the number of iterations needed. If your method fails, offer a possible reason why.
Hint: to validate your approach, compare your solution with the one obtained by doing `A\b`.