1. (a) Derive, mathematically, the back substitution method for solving the upper triangular linear system

\[ Ux = b \]

using componentwise, row-oriented and column-oriented algorithms, respectively.

(b) Write three MATLAB functions:

\[
\begin{align*}
    x &= \text{mybscomponent}(U,b) \\
    x &= \text{mybsrow}(U,b) \\
    x &= \text{mybscolumn}(U,b)
\end{align*}
\]

for the componentwise, row-oriented and column-oriented algorithms, respectively.

(c) Test the correctness of your functions and compare the execution time of these functions for a set of different sizes of upper triangular linear systems.

2. Modify the \texttt{lutx} function to a new function called \texttt{mylutx} so that it uses explicit \texttt{for} loops instead of MATLAB vector notation. For example, one section of your modified program will read

\[
\begin{align*}
    &\% \text{ compute the multipliers} \\
    &\text{for } i = k+1:n \\
    &\quad A(i,k) = A(i,k)/A(k,k); \\
    &\end{align*}
\]

Test the correctness of your function \texttt{mylutx}, and compare the execution time of \texttt{mylutx} with \texttt{lutx} and with the built-in \texttt{lu} function by finding the order of the matrix for which each of the three programs takes about 10 seconds on your computer.

3. The inverse of an \( n \times n \) matrix \( A \) can be defined as the matrix \( X \) whose columns \( x_j \) solve the equations

\[ Ax_j = e_j \quad \text{for } j = 1, 2, \ldots, n, \]

where \( e_j \) is the \( j \)th column of the identity matrix.

(a) Starting with the function \texttt{bslashx}, write a MATLAB function \( X = \text{myinv}(A) \) that computes the inverse of \( A \). Your function should call \texttt{lutx} only once and should not use the built-in MATLAB \texttt{backslash} operator or \texttt{inv} function.

(b) Test your function by comparing the inverses it computes with the inverses obtained from the built-in \texttt{inv(A)} on a few test matrices, say from “gallery” collection in MATLAB.