Math 343H Lab 5: The LU Decomposition

Objective

Last week, we used elementary matrices to reduce a matrix into row echelon form (REF). In this lab, we will discuss the LU Decomposition, which is used by numerical linear algebra packages, like MATLAB, to solve linear systems, invert matrices, and compute determinants.

The LU Decomposition

Consider the matrix from the previous lab. By iteratively left multiplying by elementary matrices, we reduce as follows:

>> A = \[4 5 6 3; 2 4 6 4; 7 8 0 5\]

>> E1 = type3m(3,2,1,-A(2,1)/A(1,1))

>> B = E1 * A

>> E2 = type3m(3,3,1,-B(3,1)/B(1,1))

>> C = E2 * B

>> E3 = \text{type3m}(3,3,2,-C(3,2)/C(2,2))

>> U = E3 * C

Note that we have reduced the above matrix into upper-triangular form, denoted as $U$. Hence, we have

$$U = E_3 E_2 E_1 A.$$ 

Since the elementary matrices are invertible, we also have

$$(E_3 E_2 E_1)^{-1} U = A.$$
This can be re-written as

\[ E_1^{-1}E_2^{-1}E_3^{-1}U = A. \]

Then we define \( L \) to be

\[ L = E_1^{-1}E_2^{-1}E_3^{-1}, \]

which yields \( LU = A \). We carry this out in MATLAB:

\[
\begin{align*}
\text{>> } L & = \text{inv(E1)} \ast \text{inv(E2)} \ast \text{inv(E3)} \\
\text{>> } L \ast U & = A
\end{align*}
\]

What makes \( LU \) decomposition so easy is that the inverses of elementary matrices are elementary matrices. For example, the inverse of a Type 3 elementary matrix is the same matrix with the opposite sign in the \((j,k)\) entry. In the above problem, we have:

\[
\begin{align*}
\text{>> } L & = \text{type3m(3,2,1,A(2,1)/A(1,1))} \ast \ldots \\
& \quad \ast \text{type3m(3,3,2,C(3,2)/C(2,2))}
\end{align*}
\]

Note that the minus signs are gone. Doing it this way, we don’t have to actually invert anything to compute \( L \). This makes the computation much faster.

**Why Should I Care?**

The LU Decomposition isn’t very useful when doing matrix computation by hand. It is, however, very important in scientific computation for the following reasons:

- If you want to solve the matrix equation \( Ax = b \), for several different \( b \)'s, you can replace \( A \) with \( L \) and \( U \), giving \( LUx = b \). Then solve the equations \( Ly = b \) and \( Ux = y \), using forward and backward substitution, respectively. This is actually faster than solving them with row reduction.

- The \( LU \) decomposition is also fast for finding both inverses and determinants.
• For really large matrices, think giga-bytes, the $LU$ decomposition is crucial. Indeed one can perform the $LU$ decomposition on a given matrix $A$ without needing additional space, that is, the program actually over-writes $A$ with $L$ and $U$. Note that since the diagonal of $L$ are all ones, they don’t need to be stored, and so the upper diagonal (including the diagonal) is $U$ and the lower diagonal (not including the diagonal) is $L$.

**Assignment**

**Problem 1.** Write a MATLAB function called `mylu`, which takes as input a random $n \times n$ matrix, performs the LU decomposition, and returns $L$ and $U$. To verify that it works, multiply $L$ and $U$ back together again to get $A$.

**Problem 2.** Write a MATLAB function called `mydet`, which uses `mylu` to find the determinant of $A$. 