1. Gaussian elimination = LU factorization

\[ A = LU. \]

where \( L \) is a unit lower triangular matrix and \( U \) a upper triangular matrix.

2. Not all matrices have the LU factorization. For example,

\[ A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 1 \\ 2 & 3 & 1 \end{bmatrix} \neq LU. \]

3. A permutation matrix \( P \) is an identity matrix with permuted rows.

Let \( P, P_1, P_2 \) be \( n \times n \) permutation matrices, and \( X \) be an \( n \times n \) matrix. Then

- \( P^T P = I \), i.e., \( P^{-1} = P^T \).
- \( \det(P) = \pm 1 \).
- \( P_1P_2 \) is also a permutation matrix.
- \( PX \) is the same as \( X \) with its rows permuted.
- \( XP \) is the same as \( X \) with its columns permuted.
- \( P_1XP_2 \) reorders both rows and columns of \( X \).

4. The need of pivoting, mathematically

The LU factorization can fail on nonsingular matrices, see the above example. But by exchanging the first and third rows, we get

\[
PA = \begin{bmatrix} 2 & 3 & 1 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 & 1 \\ 0 & 1 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} = LU.
\]

5. The above simple observation is the basis for LU factorization with pivoting.

**Theorem.** If \( A \) is nonsingular, then there exist permutations \( P \), a unit lower triangular matrix \( L \), and a nonsingular upper triangular matrix \( U \) such that

\[ PA = LU. \]

6. Function `lutx.m`

7. Solving \( Ax = b \) using the LU factorization

1. Factorize \( A \) into \( PA = LU \)
2. Permute the entries of \( b \): \( b := Pb \).
3. Solve \( L(Ux) = b \) for \( Ux \) by forward substitution:
   \[ Ux = L^{-1}b. \]
4. Solve \( Ux = L^{-1}b \) for \( x \) by back substitution:
   \[ x = U^{-1}(L^{-1}b). \]
Let us apply LU factorization without pivoting to

\[ A = \begin{bmatrix} .0001 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 10^{-4} & 1 \\ 1 & 1 \end{bmatrix} = LU = \begin{bmatrix} 1 \\ l_{21} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ u_{22} \end{bmatrix} \]

in three decimal-digit floating point arithmetic. We obtain

\[ L = \begin{bmatrix} 1 \\ fl(1/10^{-4}) & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 10^4 1 \end{bmatrix}, \]

\[ U = \begin{bmatrix} 10^{-4} \\ fl(1 - 10^4 \cdot 1) \end{bmatrix} = \begin{bmatrix} 10^{-4} & 1 \\ 1 & -10^4 \end{bmatrix}, \]

so

\[ LU = \begin{bmatrix} 1 & 0 \\ 10^4 & 1 \end{bmatrix} \begin{bmatrix} 10^{-4} & 1 \\ -10^4 & 1 \end{bmatrix} = \begin{bmatrix} 10^{-4} & 1 \\ 1 & 0 \end{bmatrix} \neq A, \]

where the original \( a_{22} \) has been entirely “lost” from the computation by subtracting \( 10^4 \) from it. In fact, we would have gotten the same LU factors whether \( a_{22} \) had been \( 1, 0, -2 \), or any number such that \( fl(a_{22} - 10^4) = -10^4 \). Since the algorithm proceeds to work only with \( L \) and \( U \), it will get the same answer for all these different \( a_{22} \), which correspond to completely different \( A \) and so completely different \( x = A^{-1}b \); there is no way to guarantee an accurate answer. This is called numerical instability. \( L \) and \( U \) are not the exact factors of a matrix close to \( A \).

Let us see what happens when we go on to solve \( Ax = [1, 2]^T \) for \( x \) using this LU factorization. The correct answer is \( x \approx [1, 1]^T \). Instead we get the following. Solving

\[ Ly = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \Rightarrow y_1 = fl(1/1) = 1 \text{ and } y_2 = fl(2 - 10^4 \cdot 1) = -10^4. \]

Note that the value 2 has been “lost” by subtracting \( 10^4 \) from it. Solving

\[ Ux = y = \begin{bmatrix} 1 \\ -10^4 \end{bmatrix} \Rightarrow \hat{x}_2 = fl((-10^4)/(-10^4)) = 1 \text{ and } \hat{x}_1 = fl((1 - 1)/10^{-4}) = 0, \]

a completely erroneous solution.

On the other hand, the LU factorization with partial pivoting would have reversed the order of the two equations before proceeding. You can confirm that we get

\[ PA = LU, \]

where

\[ P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad L = \begin{bmatrix} 1 \\ fl(.0001/1) \end{bmatrix} = \begin{bmatrix} 1 \\ .0001 \end{bmatrix}, \]

and

\[ U = \begin{bmatrix} 1 \\ fl(1 - .0001 \cdot 1) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \]

The computed LU approximates \( A \) very accurately. As a result, the computed solution \( x \) is also perfect!