A brief note on LU decomposition

Consider a square matrix $[A]_{N\times N}$ and the following equation that we are trying to solve,

$$A \cdot x = b \implies \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0N} \\ a_{10} & a_{11} & \dots & a_{1N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N0} & a_{N1} & \dots & a_{NN} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_N \end{pmatrix} \tag{1}$$

We would like to decompose $[A]$ into lower $[L]$ and upper $[U]$ matrices such that

$$A = L \cdot U \;\text{ where }\; L = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{10} & 1 & 0 & \dots & 0 \\ l_{20} & l_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{N0} & l_{N1} & l_{N2} & \dots & 1 \end{pmatrix} \;\text{ and }\; \begin{pmatrix} u_{00} & u_{01} & u_{02} & \dots & u_{0N} \\ 0 & u_{11} & u_{12} & \dots & u_{1N} \\ 0 & 0 & u_{22} & \dots & u_{2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & u_{NN} \end{pmatrix} \tag{2}$$

This implies that in order to solve x in Eqn. (1) we do the following,

$$A \cdot x = b \;\Rightarrow\; L \cdot U \cdot x = b \;\Rightarrow\; \text{if } L \cdot y = b \text{ then } U \cdot x = y \tag{3}$$

Thus we solve for x in two steps as shown in Eqn. (3), first solve y using $L \cdot y = b$ and next $U \cdot x = y$. We can factorize A in L and U either by Gauss-Jordan elimination or by so-called Crout's algorithm. Both methods require $a_{00} \neq 0$ which needs partial pivoting (row pivoting) if necessary. For numerical implementation, we discuss only Crout's algorithm here.

Consider the matrix multiplication $L \cdot U$ using Eqn. (2), and for simplification let us confine ourselves to $N = 4$.

$$L \cdot U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ l_{10} & 1 & 0 & 0 \\ l_{20} & l_{21} & 1 & 0 \\ l_{30} & l_{31} & l_{32} & 1 \end{pmatrix} \cdot \begin{pmatrix} u_{00} & u_{01} & u_{02} & u_{03} \\ 0 & u_{11} & u_{12} & u_{13} \\ 0 & 0 & u_{22} & u_{23} \\ 0 & 0 & 0 & u_{33} \end{pmatrix} \tag{4}$$

The matrix multiplication yields

$$\begin{pmatrix} u_{00} & u_{01} & u_{02} & u_{03} \\ l_{10}u_{00} & l_{10}u_{01}+u_{11} & l_{10}u_{02}+u_{12} & l_{10}u_{03}+u_{13} \\ l_{20}u_{00} & l_{20}u_{01}+l_{21}u_{11} & l_{20}u_{02}+l_{21}u_{12}+u_{22} & l_{20}u_{03}+l_{21}u_{13}+u_{23} \\ l_{30}u_{00} & l_{30}u_{01}+l_{31}u_{11} & l_{30}u_{02}+l_{31}u_{12}+l_{32}u_{22} & l_{30}u_{03}+l_{31}u_{13}+l_{32}u_{23}+u_{33} \end{pmatrix} \tag{5}$$

Equating this expression in (5) element by element to A in (1), writing the diagonal elements first,

$$a_{00} = u_{00} \qquad \Rightarrow u_{00} = a_{00} \tag{6}$$

$$a_{11} = l_{10}u_{01} + u_{11} \qquad \Rightarrow u_{11} = a_{11} - l_{10}u_{01} \tag{7}$$

$$a_{22} = l_{20}u_{02} + l_{21}u_{12} + u_{22} \qquad \Rightarrow u_{22} = a_{22} - l_{20}u_{02} - l_{21}u_{12} \tag{8}$$

$$a_{33} = l_{30}u_{03} + l_{31}u_{13} + l_{32}u_{23} + u_{33} \qquad \Rightarrow u_{33} = a_{33} - l_{30}u_{03} - l_{31}u_{13} - l_{32}u_{23} \tag{9}$$

Generically, for the diagonal elements we can write

$$\text{for } \; i = j: \qquad u_{ij} = a_{ij} - \sum_{k=0}^{i-1} l_{ik}u_{kj} \tag{10}$$

Let us consider the upper triangle *i.e.* elements above the diagonal for which $i < j$,

$$a_{01} = u_{01} \qquad \Rightarrow u_{01} = a_{01} \tag{11}$$

$$a_{02} = u_{02} \qquad \Rightarrow u_{02} = a_{02} \tag{12}$$

$$a_{03} = u_{03} \qquad \Rightarrow u_{03} = a_{03} \tag{13}$$

$$a_{12} = l_{10}u_{02} + u_{12} \qquad \Rightarrow u_{12} = a_{12} - l_{10}u_{02} \tag{14}$$

$$a_{13} = l_{10}u_{03} + u_{13} \qquad \Rightarrow u_{13} = a_{13} - l_{10}u_{03} \tag{15}$$

$$a_{23} = l_{20}u_{03} + l_{21}u_{13} + u_{23} \qquad \Rightarrow u_{23} = a_{23} - l_{20}u_{03} - l_{21}u_{13} \tag{16}$$

Generic form for u_{ij} therefore is

$$\text{for } \; i < j: \qquad u_{ij} = a_{ij} - \sum_{k=0}^{i-1} l_{ik}u_{kj} \tag{17}$$

The Eqns. (10) and (17) look identical except for the restriction on i and j. Therefore, this expression can be nicely accommodated in a `for`-loop accompanied with an `if`-statement. Now, let us turn to the lower triangle *i.e.* elements below the diagonal for which $i > j$,

$$
\begin{aligned}
a_{10} &= l_{10}u_{00} && \Rightarrow l_{10} = a_{10}/u_{00} && (18)\\
a_{20} &= l_{20}u_{00} && \Rightarrow l_{20} = a_{20}/u_{00} && (19)\\
a_{30} &= l_{30}u_{00} && \Rightarrow l_{30} = a_{30}/u_{00} && (20)\\
a_{21} &= l_{20}u_{01} + l_{21}u_{11} && \Rightarrow l_{21} = \left[a_{21} - l_{20}u_{01}\right]/u_{11} && (21)\\
a_{31} &= l_{30}u_{01} + l_{31}u_{11} && \Rightarrow l_{31} = \left[a_{31} - l_{30}u_{01}\right]/u_{11} && (22)\\
a_{32} &= l_{30}u_{02} + l_{31}u_{12} + l_{32}u_{22} && \Rightarrow l_{32} = \left[a_{32} - l_{30}u_{02} - l_{31}u_{12}\right]/u_{22} && (23)
\end{aligned}
$$

Therefore, in short we can write the l_{ij}'s as

$$
\text{for } \; i > j : \qquad\qquad l_{ij} = \frac{1}{u_{jj}}\left[a_{ij} - \sum_{k=0}^{i-1} l_{ik}u_{kj}\right] \tag{24}
$$

Needless to mention that except for the factor $1/u_{jj}$ the term in square bracket is just what you got in (10) and (17). In order to implement the above algorithm, it is obvious you have to solve column-wise. Run the i-loop first and generate the u_{00}, l_{10}, l_{20}, l_{30} for $i = 0 - 3$ keeping $j = 0$ fixed. Next move to $j = 1$ and run through $i = 0 - 3$ and so on. Try to convince yourself why row-wise solution is not possible *i.e.* keeping i fixed and running through j.

Note, that each a_{ij} is used only once. So in order to be memory efficient, one can use the space for A matrix for L and U.

The calculation of determinant becomes trivial, it is just $\Pi_i u_{ii}$. For inverse, one doesn't gain much over Gauss-Jordan because here too you have to solve $L \cdot y = b$ for each column of unit matrix followed by solving $U \cdot x = y$.