**Bonus Assignment**

**Members:**

**Saleh Hayat(61882)**

**Bilal Amin**

**Najam Khan**

**TASK**

Our takes is to make lu\_decomposition more efficient in term of time so first we take a look on how it work then we try to make it efficient in term of time using threading.

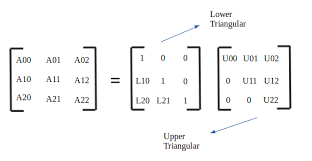
**EXPLAINING**

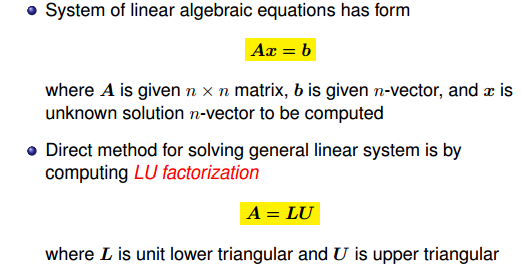
Lu\_decomposition is use to calculate some algebraic expressions which are in for of matrix

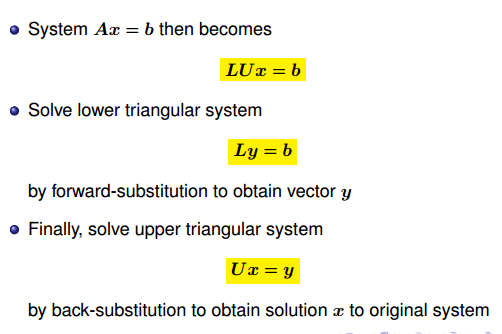
Like AX=B

A,X and B are matrixes so we decompose matrix A in L and U such that A=LU and that why its called LU decomposition

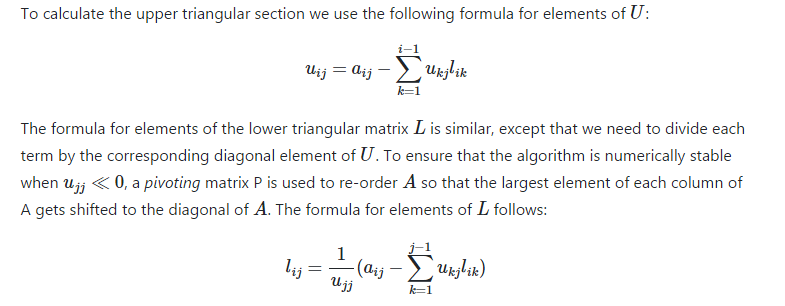
Some demonstration of content of A ,L and U



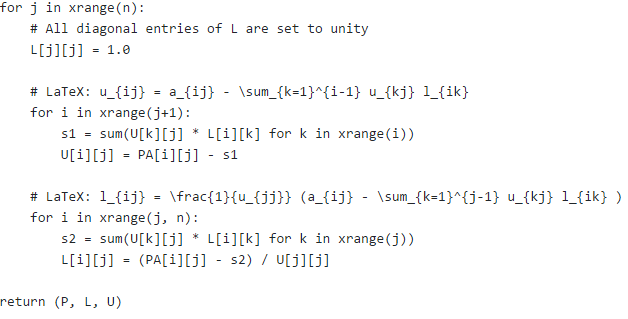




Now we have to transform the code.The most complicated and costly part of code is when we calculation values of L and U

to calculate U and L we you above formula.

So we create a loop which run equal to number of columns each column is independent of other column but rows are dependent so we can compute each column values parallel .as we can see in picture below

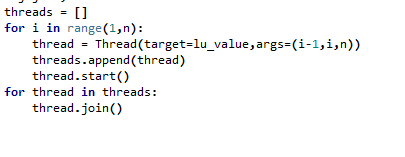


**What WE DO**

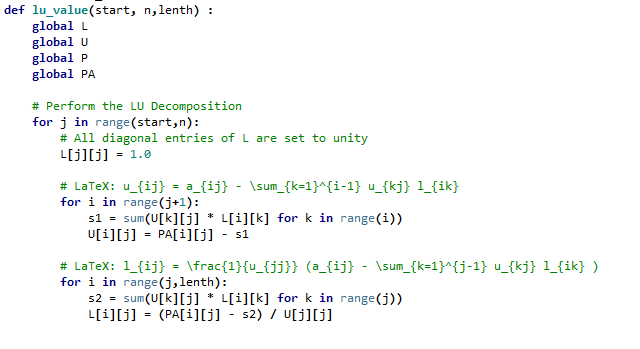
So in order to achieve our task we create threads equal to number of columns and compute them parallel

Here the code

Thread code



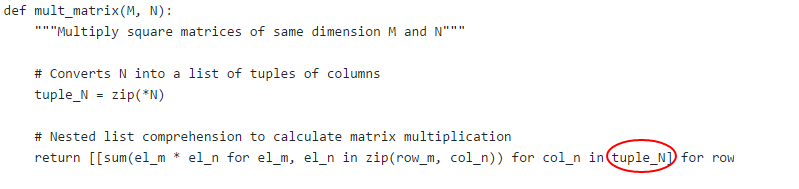
Here the method code which is executed by threads



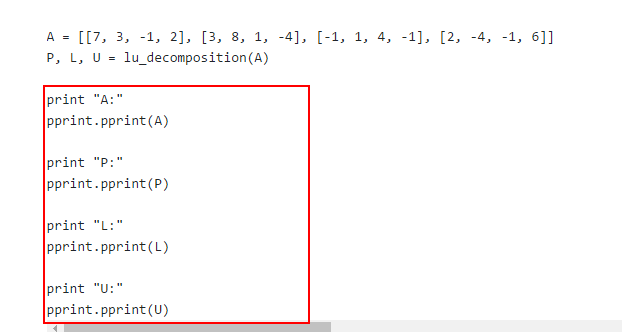
So now our the program compute the values of U and L by using thread so its much faster then before because we use thread so the computation is parallel now which make program time cost much less.

**PROBLEMS WE FACE**

The program of python which is given to us have a lot of errors . The program have error in printing and also have some logical error and we have no other document for reference so it take us to long to over come these problems. Some variables are use in wrong place as show below

they use tuple\_N instead of N

Some brackets are missing too



We over come all these problems and also it take us long because we are not familiar with python so it increases difficulty too because we are not familiar with python.

**Work Details**

Bilal Amin[All errors are tackled by bilia amin]

Najam khan[report and research is done by najam khan]

Saleh Hayat[All threading and code modification to make it parallel done by Saleh hayat]