Analyzing Matrix inversion algorithm

* Algorithm:

For calculate matrix inversion we use LUP decomposition method which describe as bellow:

For given matrix A:

LU factorization with partial pivoting as:

***PA = LU***

***L*** and ***U*** are lower and upper triangular matrices. unique factorization for matrix ***A***

require the lower triangular matrix *L* to be a unit triangular matrix.

***P*** is a permutation matrix which reorders the rows of ***A***.

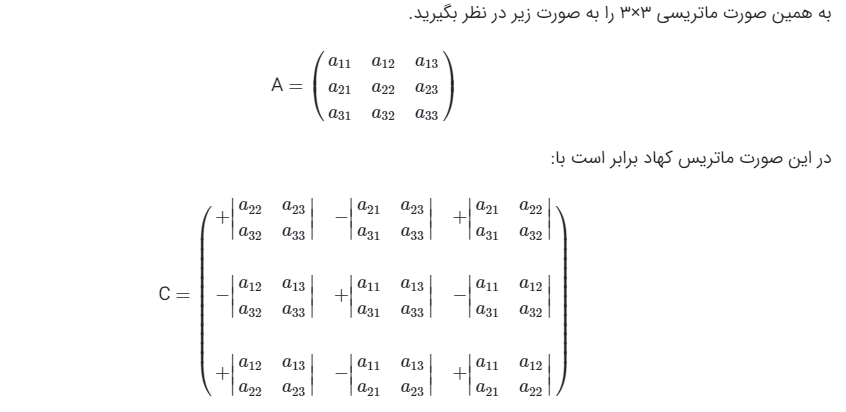
Then for calculating matrix invers we solve bellow expression in defined manner as bellow:

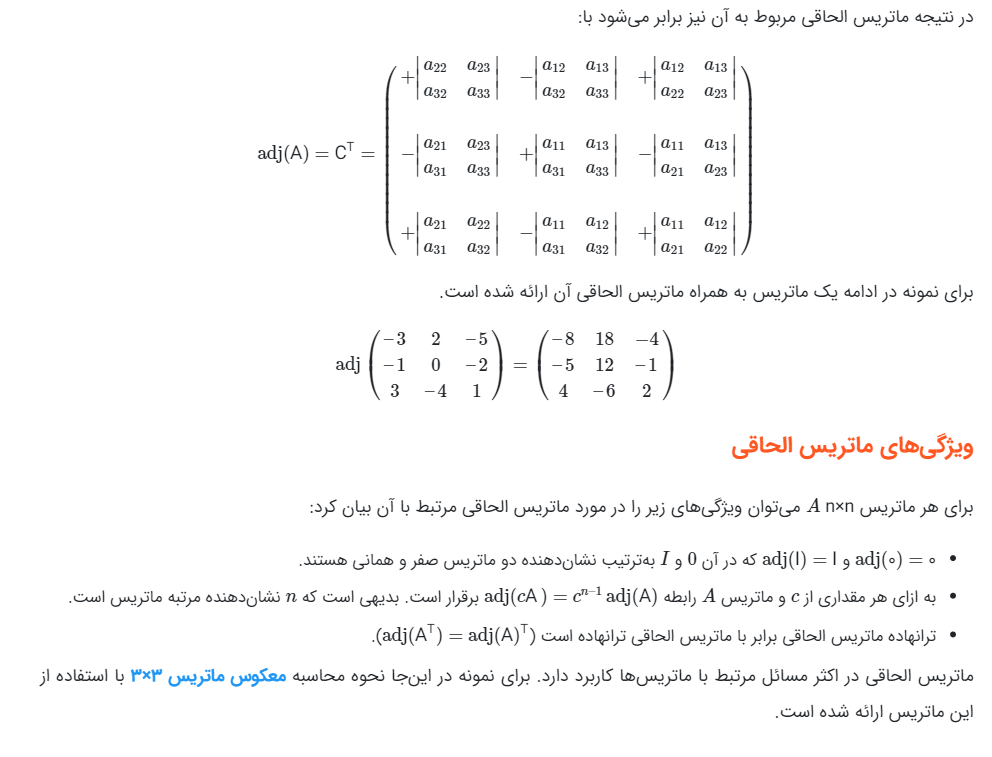
***PA = LU => AA-1 = LU A-1 = PI:***

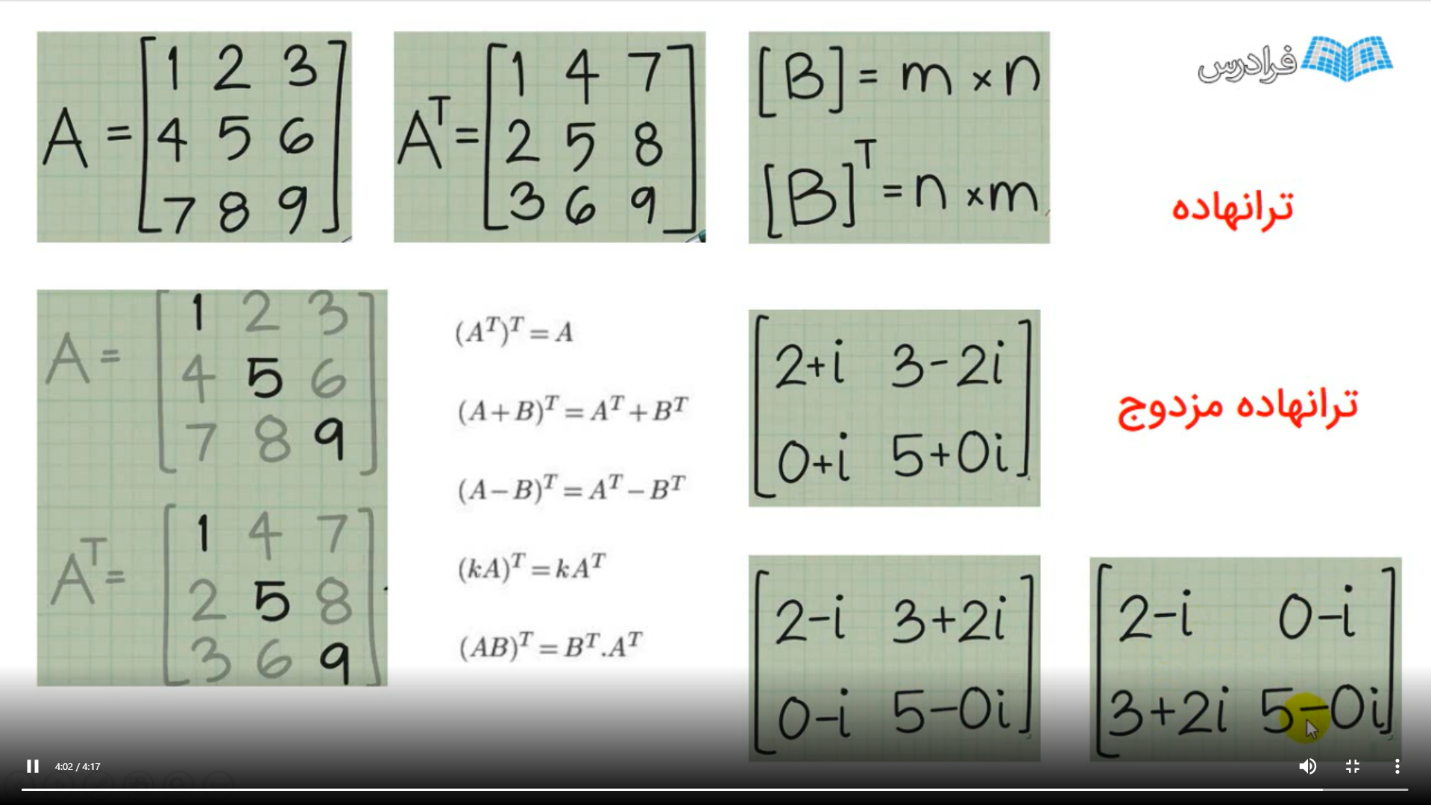
We Iteratively move over columns of I as b and solve equations:

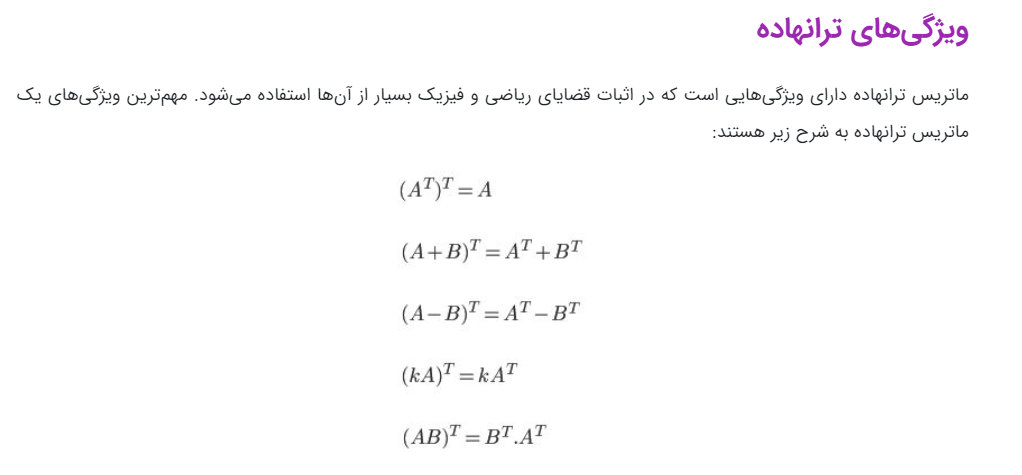
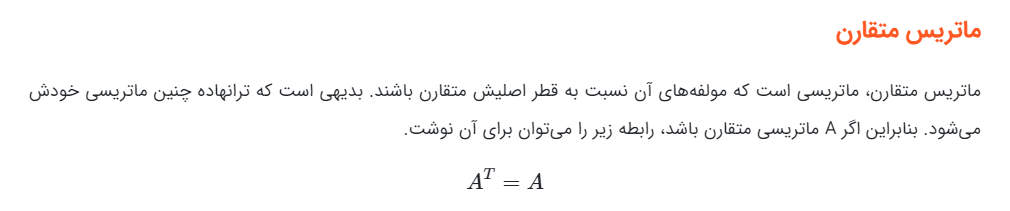
1. First, we solve the equation ***Ly = Pb*** for y.
2. Second, we solve the equation ***Ux = y*** for x.

* Implementation:

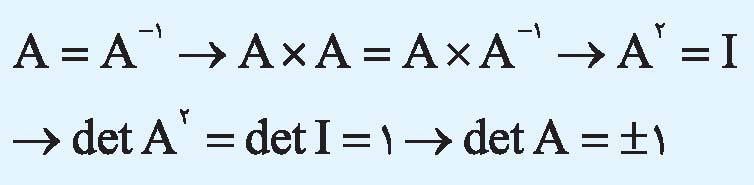


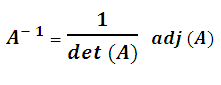






ماتریسی که تعداد سطر وستون آن با هم برابر باشند را ماتریس مربعی گویند. می‌دانیم ماتریس‌های مربعی که دترمینان غیرصفر دارند معکوس‌پذیرهستند

بنابراین اولین شرط این است که دترمینان ماتریس 1 یا 1- باشد.



Determinant of a Matrix is a special number that is defined only for square matrices

Computing an LU decomposition using either of these algorithms requires  {\textstyle {\frac {2}{3}}n^{3}}{\textstyle {\frac {2}{3}}n^{3}}2/3 \* n^3  floating-point operations

**Theoretical complexity**

 LU decomposition can be computed in time O(*M*(*n*)). *M*(*n*) ≥ *na where a > 2.*

*It means* O(*n*2.376)

Lu decomposition use partial pivoting.

Partial Pivoting

place the largest entry of the first

column of the matrix at the top of that first column. Find largest need O(n2) comparisons.

You're probably just getting a stack overflow here. The array is too big to fit in your program's stack address space.

If you allocate the array on the heap you should be fine, assuming your machine has enough memory.

int\* array = new int[1000000];

gcc parameters used:

-O2 : optimize more

-O3 : optimize even more

'-Wformat' which is included in '-Wall'.

Check calls to printf and scanf, etc., to make sure that the arguments supplied have types appropriate to the format string specified, and that the conversions specified in the format string make sense.

'-fgcse-lm'

When '-fgcse-lm' is enabled, global common subexpression elimination will attempt to move loads which are only killed by stores into themselves.

Parameters in C functions

Pass by Value:

copy of the data passed, changes have no effect outside.

Pass by Reference:

"refers" to the original data passed, changes also made to the original data.

For implementing this method:

C++ use & void

void function( int & x ){}

call : function(x);

Standard C - Using "Pointers":

void function( int \* x ){}

call : function(&x);

Arrays are always **passed by reference** in C. They **do not** use the '&' notation.

Specifying stack size by gcc:

(assuming Windows) you can try passing

|  |
| --- |
| -Wl,--stack,<size> |

(where <size> is in bytes)  
to set the stack size.  
  
(From doing some tests, it seems the default is 2 MB)

With stack size as 2mg as bellow:

gcc -Wl,--stack,2194304 lup\_matrix\_inverse.c -o out.exe

couldn’t run for size 500 due to segment falut.

It solved by using 4mg stack size as bellow:

gcc -Wl,--stack,4194304 lup\_matrix\_inverse.c -o out.exe