**Problem Statement**
How much computational time does it take to conduct back substitution in the LU Decomposition method?

**Solution**
At the beginning of back substitution, the set of equations is of the form

$$[U][X]=[B]$$

where

$[U]=$ upper triangular matrix, $n\times n$,
$[X]=$ unknown vector, $n\times 1$, and
$[B]=$ right hand side vector, $n\times 1$.

The algorithm for finding the unknown vector is

$$x_n=\frac{b_n}{u_{nn}} \qquad (1)$$

$$x_i=\frac{b_i-\sum_{j=i+1}^{n}u_{ij}x_j}{u_{ii}},\ i=n-1,n-2,\ldots,2,1 \qquad (2)$$

**Hint 1**
For an arithmetic progression series summation

$$S=a_1+a_2+\ldots+a_{m-1}+a_m$$
$$=\frac{m}{2}(a_1+a_m)\ \text{or}\ \frac{m}{2}(2a_1+(m-1)d)$$

where

$$d=a_{i+1}-a_i, i=1,2..,m-1$$

**Hint2**
When you add $n$ terms together, there are $n$-1 additions.

Now let us see how many arithmetic operations are needed to complete the algorithm.

**Number of Divisions**
See equation (1). We have one operation of division for the calculation of $x_n$.
See equation (2). For each $x_i$, $i=n-1,n-2,\ldots,2,1$, the result is divided once by $u_{ii}$.
So total number of divisions is 1+($n$-1) =$n$

**Number of Subtractions**
See equation (2). For each $x_i$, $i=n-1,n-2,\ldots,2,1$, we have 1 subtraction in the numerator.
So total number of subtractions is $n$-1.

**Number of Multiplications**

See equation (2). For a particular $i$, there are $(n-i)$ multiplications – just count the $u_{ij}x_j$ terms within the summation in the numerator. Since $i$ takes the values of $i = n-1, n-2, \ldots, 2, 1$, you can see that

when $i$=$n$-1, there is ($n$-($n$-1))=1 multiplication
when $i$=$n$-2, there are ($n$-($n$-2))=2 multiplications
.
.
.
when $i$=1, there are ($n$-1)=$n$-1 multiplications
So the total number of multiplications is

$$1+2+\ldots+(n-1)$$
$$=\frac{n-1}{2}(1+(n-1))$$
$$=\frac{n(n-1)}{2}$$

**Number of Additions**

For a particular $i$, there are $(n-i-1)$ additions – just see the number of $u_{ij}x_j$ terms within the summation in the numerator of equation (2).
Since $i$ takes the values of $i = n-1, n-2, \ldots, 2, 1$, you can see that
when $i$=$n$-1, there is ($n$-($n$-1)-1))=0 addition
when $i$=$n$-2, there is ($n$-($n$-2)-1))=1 addition
.
.
.
when $i$=1, there are ($n$-(1)-1))=$n$-2 additions
So total number of additions is

$$0+1+2+\ldots+(n-2)$$
$$=\frac{n-1}{2}(0+(n-2))$$
$$=\frac{(n-1)(n-2)}{2}$$

**Computational Time**

Assuming it takes 4 clock cycles for each addition, subtraction, and multiplication, and 16 clock cycles for each division, then if $C$ is the clock cycle time,

computational time spent on divisions= $n\times(16C)=16Cn$

computational time spent on subtractions $=(n-1)\times(4C)=4C(n-1)$

computational time spent on multiplications $=\frac{n(n-1)}{2}\times(4C)=2Cn(n-1)$

computational time spent on additions $= \frac{(n-1)(n-2)}{2} \times (4C) = 2C(n-1)(n-2)$

The total computational time spent on back substitution then is

=Total time for divisions + Total time for subtractions + Total time for multiplications + Total time for additions

$$= 16Cn + 4C(n-1) + 2Cn(n-1) + 2C(n-1)(n-2)$$

$$= 16Cn + 4Cn - 4C + 2Cn^2 - 2Cn + 2Cn^2 - 6Cn + 4C$$

$$= 4Cn^2 + 12Cn$$

$$= C(4n^2 + 12n)$$

## Questions

1. How much computational time does it take to conduct forward substitution?
2. How much computational time does it take to conduct forward elimination?
3. How much computational time does it take to conduct back substitutions?
4. How much times does it take to conduct Gaussian elimination?
5. How much times does it take to conduct LU decomposition?
6. How much computational time does it take to find the inverse of a square matrix using LU decomposition?
7. How much computational time does it take to find the inverse of a square matrix using Gaussian elimination?

## References

LU Decomposition
http://numericalmethods.eng.usf.edu/topics/lu_decomposition.html