Practical 3

- Solve the Ax=b for x , by back substitution *

$$A = \begin{bmatrix} 1 & 3 & 1 \\ 10 & 1 & -2 \\ -6 & -1 & 2 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 2 \\ 10 \end{bmatrix}$$

****

Method

First of all we will create a function, using *rlu()* to transform A into echlon form then we will back substitute

Workout

First of all we need to put the matrix in R, we will put the augmented matrix in A

```
rm(list=ls())
A=matrix(c(1,3,1,10,1,-2,-1,-6,2,1,2,10),nrow=3,byrow=T)
print(A)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    3    1   10
## [2,]    1   -2   -1   -6
## [3,]    2    1    2   10
```

Here creating *rlu()* to transform in row echloen form

```
rlu=function(x)
{
if(is.matrix(x)==1)
{
for(i in 2:nrow(x))
{
x[i, ]<-x[i, ]-x[i,1]*(x[1, ]/x[1,1])
}
x[-1,-1]<-rlu(x[-1,-1])
}
x
}
```

Now calculating row echloen form

```
L=rlu(A)
L=round(L,4)
print(L)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    3    1   10
## [2,]    0   -5   -2  -16
## [3,]    0    0    2    6
```

Now taking A and b separated

```r
A=L[ ,1:3]
b=L[ ,4]
print(A)
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    1
## [2,]    0   -5   -2
## [3,]    0    0    2
```

```r
print(b)
```

```
## [1]  10 -16   6
```

Checking Backsubstituting

```r
  n=dim(A)[1]
  p=dim(A)[2]
  b=as.matrix(b)
  for(j in p:1)
  {
    b[j,1]=b[j,1]/A[j,j]

    if((j-1)>1)
      b[(j-1),1]= b[(j-1),1]-(b[j,1]*A[(j-1),j])
    else
      b[(j-1),1]= b[(j-1),1]-((b[j,1]*A[(j-1),j])+(b[(j+1),(j-1)]*A[(j-1),(j+1)]))
  }
  print(b)
```

```
##      [,1]
## [1,]    1
## [2,]    2
## [3,]    3
```

Conclusion

So A ,U ,L are

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$