

Assignment 2

June 2, 2019

In []:

```
In [61]: import numpy as np
import numpy as np
np.set_printoptions(threshold=np.inf)
import scipy
from scipy.linalg import lu
ROW = 3
COL = 3
#np.random.seed(0)
matrix = np.random.randint(low=0,high=2,size=(ROW,COL) )
print (matrix)
```

```
[[1 0 1]
 [0 1 0]
 [1 0 0]]
```

In [62]: matrix.dtype

Out[62]: dtype('int64')

```
In [63]: for k in range (0, ROW-1):
    #Step1 : Check pivot to be non-zero : else swap
    if matrix[k,k] == 0 :
        print("Diagonal element is ZERO")
        temp = matrix.copy()
        if ( k != (ROW-1) ):
            matrix[k] = temp[(k+1)]
            matrix[(k+1)] = temp[k]

    #Step2 : Make all elements below the pivot to be zero.
    for i in range (k, ROW-1):
        matrix[(i+1)] = matrix[(i+1)]*matrix[k,k] - matrix[k]*matrix[(i+1),k]
        print("k", k, "i+1", (i+1))
        print(matrix)
        print("")
```

```

#Step3 : Noramalize each row by its pivot.
matrix = matrix * 1.0
for i in range (0, ROW):
    if matrix[i,i] != 0 :
        matrix[i] = matrix[i]/matrix[i,i]

print("Normal Matrix")
print(matrix)

```

```

k 0 i+1 1
[[1 0 1]
 [0 1 0]
 [1 0 0]]

```

```

k 0 i+1 2
[[ 1  0  1]
 [ 0  1  0]
 [ 0  0 -1]]

```

```

k 1 i+1 2
[[ 1  0  1]
 [ 0  1  0]
 [ 0  0 -1]]

```

```

Normal Matrix
[[ 1.  0.  1.]
 [ 0.  1.  0.]
 [-0. -0.  1.]]

```

In [64]: *#Step4: REF to RREF by making elements above each pivot to be Zero*

```

for i in range (0, ROW):
    k = ROW - 1 -i
    for l in range (0, k):
        j = ROW -2 -i - 1
        matrix[j] = matrix[j] - (matrix[k] * matrix[j,k])
    print(matrix)
    print("")

```

```

[[ 1.  0.  1.]
 [ 0.  1.  0.]
 [-0. -0.  1.]]

```

```

[[ 1.  0.  0.]
 [ 0.  1.  0.]
 [-0. -0.  1.]]

```

```
[[ 1.  0.  0.]  
 [ 0.  1.  0.]  
 [-0. -0.  1.]]
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```