**SOLUTION OF LINEAR EQUATIONS BY GAUSS JORDAN & GAUSS-ELIMINATION METHOD**

**SCILAB ALGORITHM**

*//gauss elimination method and gauss jordan method*

clc; clear;

a=[12 3 -5; 1 5 3; 3 7 13]

b=[1;28;76]

n=length(b)

a=[a b]

for j=1:n-1 *//Gauss elimination*

for i=j+1:n *//Gauss elimination*

*//for j=1:n //Gauss jordan*

*// a(j,:)=a(j,:)/a(j,j)//Gauss jordan*

*// for i=1:n //Gauss jordan*

if i~=j

m=a(i,j)/a(j,j)

a(i,:)=a(i,:)-(m\*a(j,:))

end

end

end

disp(a)

x=zeros(1,n)

for i=n:-1:1

ax=sum(a(i,1:n).\*x)

x(i)=(a(i,n+1)-ax)/a(i,i)

disp('x('+string(i)+')='+string(x(i)))

end

scf(0)

bar(x)

xtitle("Solution of Linear System (Gauss Elimination)","Variable Index","Value")

**Python**

#gauss elimination method and gauss jordan method

import numpy as np

import matplotlib.pyplot as plt

A=np.array([[12, 3, -5], [1, 5, 3], [3, 7, 13]],dtype=float)

b=np.array([[1],[28],[76]])

n=len(b)

a=np.hstack((A, b))

for j in range(n-1): # Gauss Elimination

#for j in range(n): # Gauss Jordan

#a[j,:]=a[j,:]/a[j,j] # Gauss Jordan

#for i in range(n): # Gauss Jordan

for i in range (j+1,n): # Gauss Elimination

if i!=j:

m=a[i,j]/a[j,j]

a[i,:]=a[i,:]-(m\*a[j,:])

x=np.zeros(n)

for i in range(n-1,-1,-1):

ax=np.sum(a[i,i+1:n]\*x[i+1:n])

x[i] = (a[i, -1] - ax) / a[i, i]

print("\nSolution (Gauss Elimination):", x[i])

plt.figure(figsize=(5,4))

plt.bar(range(1, n+1), x)

plt.title("Solution of Linear System (Gauss Elimination)")

plt.xlabel("Variable Index")

plt.ylabel("Value")

plt.show()