**SOLUTION OF LINEAR EQUATIONS BY GAUSS JACOBI & GAUSS-SIEDEL METHOD**

**SCILAB ALGORITHM**

*// gauss Jacobi method and gauss siedel method*

clc;clear;

a=[3 -1 1;-1 4 1;-1 1 5]

b=[20;6;7]

n=length(b)

c=0

for i=1:n

s=0

for j=1:n

if j~=i

s=s+a(i,j)

end

end

if (a(i,i)>s)

c=c+1

else break

end

end

if c==n then

disp('matrix is diagonally dominant')

else

disp('matrix is diagonally not dominant')

break

end

x=input('Initial guess of first value ');

y=input('Initial guess of second value ');

z=input('initial guess of third value ');

n=input('Enter no. of iterations ')

for i=1:n

x(i+1)=(b(1,1)-(a(1,2)\*y(i))-(a(1,3)\*z(i)))/a(1,1)

y(i+1)=(b(2,1)-a(2,1)\*x(i+1)-a(2,3)\*z(i))/a(2,2)

z(i+1)=(b(3,1)-a(3,1)\*x(i+1)-a(3,2)\*y(i+1))/a(3,3)

if (abs(x(i+1)-(x(i)))<0.001)& (abs(y(i+1)-(y(i)))<0.001) & (abs(z(i+1)-(z(i)))<0.001)

break

end

end

disp("No. of iteration",i,x(i),y(i),z(i))

iter = 0:i

scf(0)

plot(iter, x(1:i+1), 'ro-')

plot(iter, y(1:i+1), 'gs-')

plot(iter, z(1:i+1), 'b^-')

legend("x","y","z")

xlabel("Iteration")

ylabel("Value")

title("Convergence of Gauss-Seidel Method")

xgrid()

**Python**

#Gauss-jacobi & Gauss-Siedel

import numpy as np

import matplotlib.pyplot as plt

A = np.array([[3, -1, 1],

[-1, 4, 1],

[-1, 1, 5]], dtype=float)

b = np.array([20, 6, 7], dtype=float)

n = len(b)

c = 0

for i in range(n):

s=0

for j in range(n):

if j != i:

s=s+A[i,j]

if (A[i,i]>s):

c += 1

else:

break

if c == n:

print("Matrix is diagonally dominant")

else:

print("Matrix is diagonally not dominant")

x0 = float(input("Initial guess of first value: "))

y0 = float(input("Initial guess of second value: "))

z0 = float(input("Initial guess of third value: "))

max\_iter = int(input("Enter number of iterations: "))

x\_vals = [x0]

y\_vals = [y0]

z\_vals = [z0]

for i in range(max\_iter):

x\_new = (b[0] - (A[0,1]\*y\_vals[-1]) - (A[0,2]\*z\_vals[-1])) / A[0,0]

y\_new = (b[1] - A[1,0]\*x\_new - A[1,2]\*z\_vals[-1]) / A[1,1]

z\_new = (b[2] - A[2,0]\*x\_new - A[2,1]\*y\_new) / A[2,2]

x\_vals.append(x\_new)

y\_vals.append(y\_new)

z\_vals.append(z\_new)

if (abs(x\_vals[-1] - x\_vals[-2]) < 0.001 and

abs(y\_vals[-1] - y\_vals[-2]) < 0.001 and

abs(z\_vals[-1] - z\_vals[-2]) < 0.001):

break

print(f"No. of iterations = {i+1}")

print(f"x = {x\_vals[-1]:.4f}, y = {y\_vals[-1]:.4f}, z = {z\_vals[-1]:.4f}")

iterations = list(range(len(x\_vals)))

plt.figure(figsize=(8,6))

plt.plot(iterations, x\_vals, marker='o', label='x')

plt.plot(iterations, y\_vals, marker='s', label='y')

plt.plot(iterations, z\_vals, marker='^', label='z')

plt.xlabel("Iteration")

plt.ylabel("Value")

plt.title("Convergence of Gauss-Seidel Method")

plt.legend()

plt.grid(True)

plt.show()