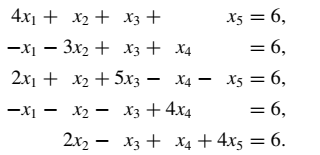
**Program to Solve by Jacobi’s method and the Guass- Seidal method and display the code and the output for the following Linear System**

**(10 Iterations)**



**Jacobi:**

**Code:**

problem = [[4.0, 1.0, 1.0, 0.0, 1.0, 6.0], #The system given in the statement is presented as a 5x6 matrix

[-1.0, -3.0, 1.0, 1.0, 0.0, 6.0],

[2.0, 1.0, 5.0, -1.0, -1.0, 6.0],

[-1.0, -1.0, -1.0, 4.0, 0.0, 6.0],

[0.0, 2.0, -1.0, 1.0, 4.0, 6.0]]

def jacobi():

x = [0.0, 0.0, 0.0, 0.0, 0.0] #previous iteration solution

nextx = [0.0, 0.0, 0.0, 0.0, 0.0] #new solution

iteration = 0

for i in range(5): #These loops take all terms on LHS to RHS (except for leading term)

for j in range(5): #These loops also divide all terms by the coefficient of the leading term

if i != j:

problem[i][j] \*= -1

problem[i][j] /= problem[i][i]

problem[i][5] /= problem[i][i]

problem[i][i] = 0

for q in range(11): #These loops print the solution of previous iteration

#These loops also generate solution for new iteration

print("Solution of system in iteration number ", iteration, ": ")

for i in range(5):

print("X", i + 1, " : ", x[i])

for i in range(5):

tempsum = 0.0

for j in range(5):

tempsum += (problem[i][j] \* x[j])

tempsum += problem[i][5]

nextx[i] = tempsum

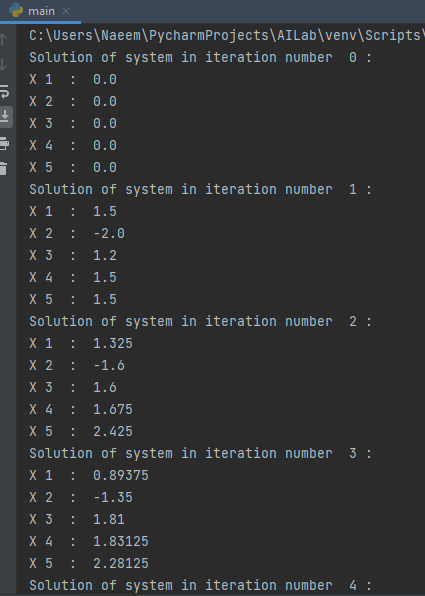
for i in range(5):

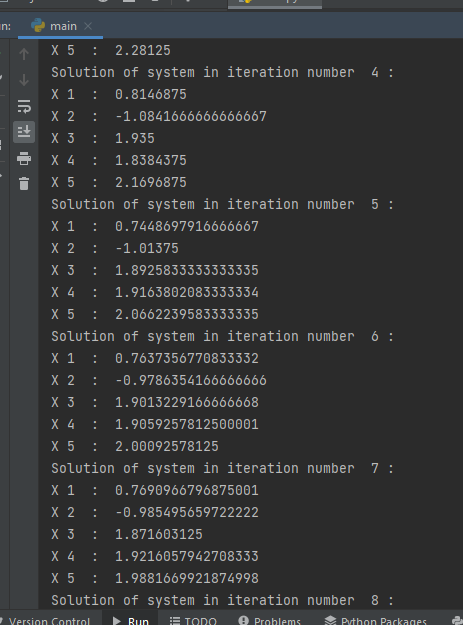
x[i] = nextx[i]

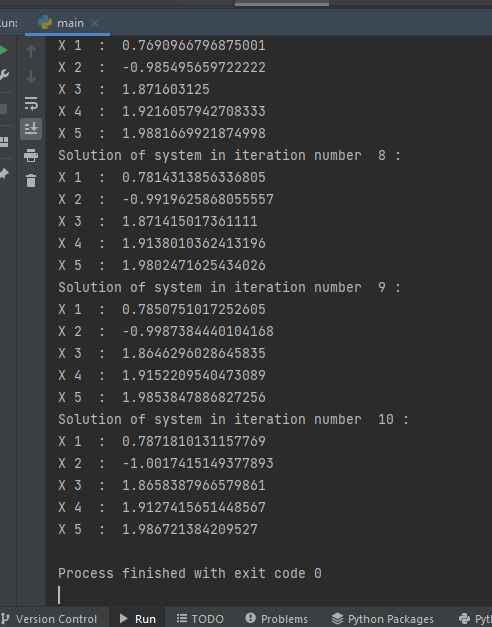
iteration += 1

jacobi()

**Screenshots:**

****

****

****

**Gauss Seidel:**

**Code:**

problem = [[4.0, 1.0, 1.0, 0.0, 1.0, 6.0], #The system given in the statement is presenated as a 5x6 matrix

[-1.0, -3.0, 1.0, 1.0, 0.0, 6.0],

[2.0, 1.0, 5.0, -1.0, -1.0, 6.0],

[-1.0, -1.0, -1.0, 4.0, 0.0, 6.0],

[0.0, 2.0, -1.0, 1.0, 4.0, 6.0]]

def gaussSeidel():

x = [0.0, 0.0, 0.0, 0.0, 0.0] #previous iteration solution

iteration = 0

for i in range(5): #These loops take all terms on LHS to RHS (except for leading term)

for j in range(5): #These loops also divide all terms by the coefficient of the leading term

if i != j:

problem[i][j] \*= -1

problem[i][j] /= problem[i][i]

problem[i][5] /= problem[i][i]

problem[i][i] = 0

for q in range(11): #These loops print the solution of previous iteration

#These loops also generate solution for new iteration

print("Solution of system in iteration number ", iteration, ": ")

for i in range(5):

print("X", i + 1, " : ", x[i])

for i in range(5):

tempsum = 0.0

for j in range(5):

tempsum += (problem[i][j] \* x[j])

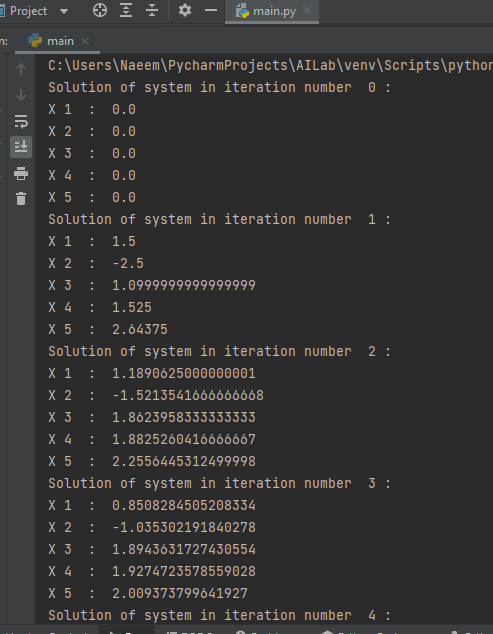
tempsum += problem[i][5]

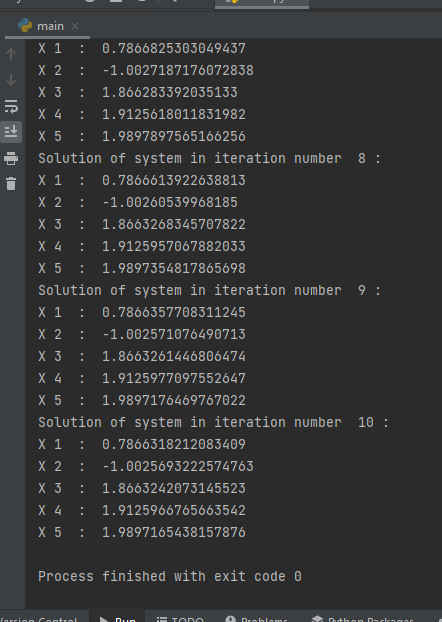
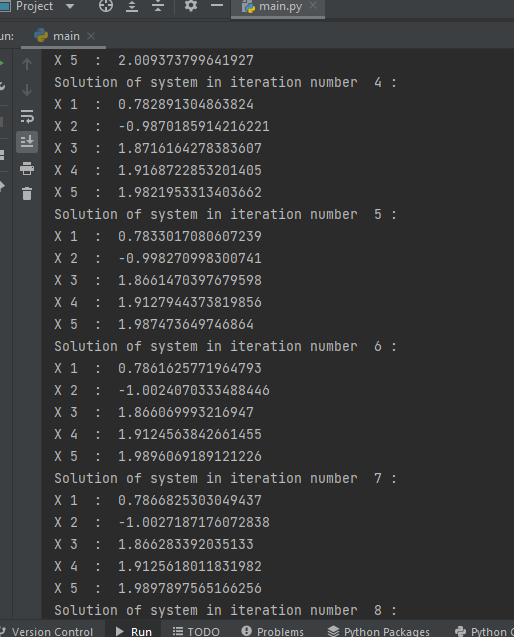
x[i] = tempsum

iteration += 1

gaussSeidel()

**Screenshots:**

****

****