# 采用雅克比迭代法求解线性代数方程组

import numpy as np

def Jacobi\_solve(A,b,x0,tol,maxit):

    n = len(b)

    x = np.zeros(n)

    error = 1

    k = 0

    print(' k                              x^k                                err')

    while (error>tol and k <= maxit):

        for i in range(n):

            s = 0

            for j in range(i):

                s += A[i,j]\*x0[j]

            for j in range(i+1,n):

                s += A[i,j]\*x0[j]

            x[i] = (b[i]-s)/A[i,i]

        error = max(abs(x-x0))

        k += 1

        x0 = np.array(x)

        print('{:2}'.format(k),'{:.15f}'.format(x[0]),'{:.15f}'.format(x[1]),'{:.15f}'.format(x[2]),'{:.15f}'.format(error))

    return x,error,k

# test example

A  = np.array([[8,-3,2],[4,11,-1],[6,3,12]])

b  = np.array([20,33,36])

x0 = np.array([0,0,0])

x, err, k=Jacobi\_solve(A, b, x0,1.0e-15,20)