

Higher Institute of Engineering & Technology, El-Beheira

Computer Engineering Department

Third assignment in numerical analysis

(The numerical solution of a system of linear equations using The Jacobi Method and The Gauss-Seidel Method.)

Under supervision of Dr.Mahmoud Gamal

|  |  |
| --- | --- |
| Team | ID |
| Mohamed Yosry El-Zarka | 19100 |
| Youssef Mohamed El-Sheheimy | 19124 |
| Omar Abd Al-Halim Khalil | 19138 |

**The Jacobi Method Source code in python: -**

1 def calc\_table\_element(i,matrix\_row,last\_x):

2     res=0.0

3     res+=matrix\_row[-1]

4

5     for j in range(0,len(matrix\_row)-1):

6         if j==i: continue

7         res-=matrix\_row[j]\*last\_x[j]

8

9     return res/matrix\_row[i]

10

11 def check\_equalization(last\_x,next\_x):

12     for i in range (0, len(last\_x)):

13         if last\_x[i] != next\_x[i]:

14             return False

15     return True

16

17 print('Project for "Numerical analysis". under the supervision of Dr. Mahmoud Gamal')

18 print('by:')

19 print('\t\tMohamed Yosry ElZarka 19100')

20 print('\t\tYoussef Mohamed Elsheheimy 19124')

21 print('\t\tOmar Abd Al-Halim Khalil 19138\n')

22

23 print("This is a program to calculate the numerical solution of a system of linear equations using The Jacobi Method.\n")

24

25 while True:

26     n=int( input('Enter the number of equations: ') )

27     print("Enter Elements of each row of the AUG matrix with the dimensions of ({} Rows X {} Columns)".format(n,n+1))

28     table=[]

29     matrix=[]

30     last\_x=[]

31     next\_x=[]

32     for i in range(0,n):

33         table.append([0.0])

34         last\_x.append(0.0)

35         next\_x.append(0.0)

36         matrix.append([]) #declare n equations

37         matrix[i]=[float(item) for item in input("Enter row #{} : ".format(i+1)).split()]

38         while len(matrix[i])!=n+1:

39             print("ERROR, Please enter",n+1, "items per row")

40             matrix[i].clear

41             matrix[i]=[float(item) for item in input("Enter row #{} : ".format(i+1)).split()]

42

43     for i in range(0,n):

44         print("row #",i+1,":",matrix[i])

45

46     for iterations in range(0,200):

47         for i in range(0,n):

48             next\_x[i]=calc\_table\_element(i, matrix[i],last\_x)

49             next\_x[i]=round(next\_x[i],3)

50             table[i].append(next\_x[i])

51         if check\_equalization(last\_x,next\_x):

52             break

53         last\_x=next\_x.copy()

54

55     for i in range(0,n):

56         print ('\nx{}= '.format(i+1),table[i])

57

58     print('')

59     for i in range(0,n):

60         print('x{}= '.format(i+1),last\_x[i])

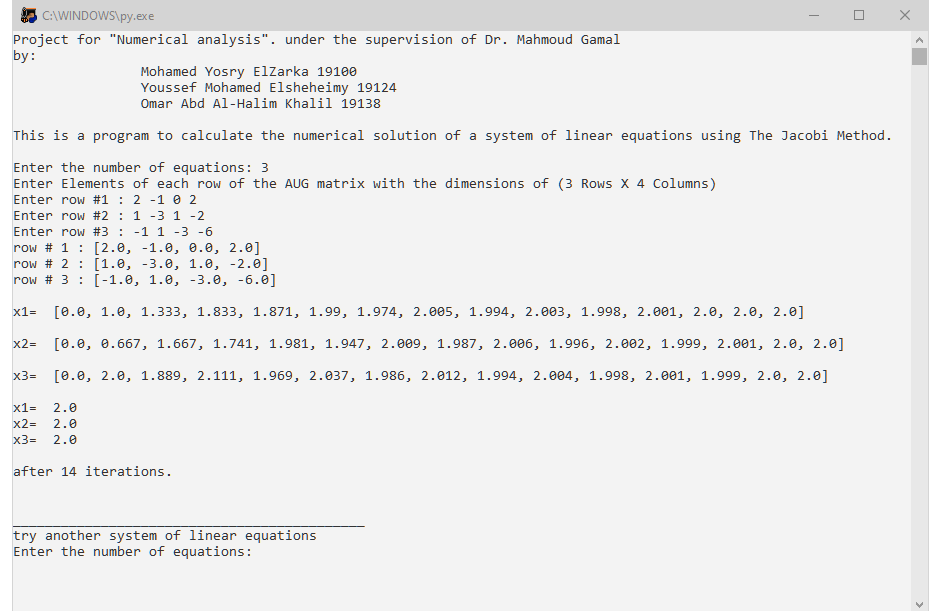
61

62     print("\nafter", iterations+1 ,"iterations.")

63     print("\n\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

64     print("try another system of linear equations")

**The Jacobi Method program**



**The Gauss-Seidel Method Source code in python: -**

1 def calc\_table\_element(i,matrix\_row,recent\_x):

2     res=0.0

3     res+=matrix\_row[-1]

4

5     for j in range(0,len(matrix\_row)-1):

6         if j==i: continue

7         res-=matrix\_row[j]\*recent\_x[j]

8

9     return res/matrix\_row[i]

10

11 def check\_equalization(recent\_x,previous\_x):

12     for i in range (0, len(recent\_x)):

13         if recent\_x[i] != previous\_x[i]:

14             return False

15     return True

16

17 print('Project for "Numerical analysis". under the supervision of Dr. Mahmoud Gamal')

18 print('by:')

19 print('\t\tMohamed Yosry ElZarka 19100')

20 print('\t\tYoussef Mohamed Elsheheimy 19124')

21 print('\t\tOmar Abd Al-Halim Khalil 19138\n')

22

23 print("This is a program to calculate the numerical solution of a system of linear equations using The Gauss-Seidel Method.\n")

24

25 while True:

26     n=int( input('Enter the number of equations: ') )

27     print("Enter Elements of each row of the AUG matrix with the dimensions of ({} Rows X {} Columns)".format(n,n+1))

28     table=[]

29     matrix=[]

30     recent\_x=[]

31     previous\_x=[]

32     for i in range(0,n):

33         table.append([0.0])

34         recent\_x.append(0.0)

35         previous\_x.append(0.0)

36         matrix.append([]) #declare n equations

37         matrix[i]=[float(item) for item in input("Enter row #{} : ".format(i+1)).split()]

38         while len(matrix[i])!=n+1:

39             print("ERROR, Please enter",n+1, "items per row")

40             matrix[i].clear

41             matrix[i]=[float(item) for item in input("Enter row #{} : ".format(i+1)).split()]

42

43     for i in range(0,n):

44         print("row #",i+1,":",matrix[i])

45

46     for iterations in range(0,200):

47         for i in range(0,n):

48             recent\_x[i]=calc\_table\_element(i, matrix[i],recent\_x)

49             recent\_x[i]=round(recent\_x[i],3)

50             table[i].append(recent\_x[i])

51         if check\_equalization(recent\_x,previous\_x):

52             break

53         previous\_x=recent\_x.copy()

54

55     for i in range(0,n):

56         print ('\nx{}= '.format(i+1),table[i])

57

58     print('')

59     for i in range(0,n):

60         print('x{}= '.format(i+1),recent\_x[i])

61

62     print("\nafter", iterations+1 ,"iterations.")

63     print("\n\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

64     print("try another system of linear equations")

**The Gauss-Seidel Method program**

