



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**

Mohamed Yosry El-Zarka  
Youssef Mohamed El-Sheheimy  
Omar Abd Al-Halim Khalil

**ID**

19100  
19124  
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
18       Dr. Mahmoud Gamal')
19 print('by:')
20 print('\t\tMohamed Yosry ElZarka 19100')
21 print('\t\tYoussef Mohamed Elsheheimy 19124')
22 print('\t\tOmar Abd Al-Halim Khalil 19138\n')
23
24 print("This is a program to calculate the numerical solution of a
25       system of linear equations using The Jacobi Method.\n")
26
27 while True:
28     n=int( input('Enter the number of equations: ') )
29     print("Enter Elements of each row of the AUG matrix with the
30           dimensions of ({} Rows X {} Columns)".format(n,n+1))
31     table=[]
32     matrix=[]
33     last_x=[]
34     next_x=[]
35     for i in range(0,n):
36         table.append([0.0])
37         last_x.append(0.0)
38         next_x.append(0.0)
39         matrix.append([]) #declare n equations
40         matrix[i]=[float(item) for item in input("Enter row #{} :
41           ".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

```
C:\WINDOWS\py.exe
Project for "Numerical analysis". under the supervision of Dr. Mahmoud Gamal
by:
    Mohamed Yosry ElZarka 19100
    Youssef Mohamed Elsheheimy 19124
    Omar Abd Al-Halim Khalil 19138

This is a program to calculate the numerical solution of a system of linear equations using The Jacobi Method.

Enter the number of equations: 3
Enter Elements of each row of the AUG matrix with the dimensions of (3 Rows X 4 Columns)
Enter row #1 : 2 -1 0 2
Enter row #2 : 1 -3 1 -2
Enter row #3 : -1 1 -3 -6
row # 1 : [2.0, -1.0, 0.0, 2.0]
row # 2 : [1.0, -3.0, 1.0, -2.0]
row # 3 : [-1.0, 1.0, -3.0, -6.0]

x1= [0.0, 1.0, 1.333, 1.833, 1.871, 1.99, 1.974, 2.005, 1.994, 2.003, 1.998, 2.001, 2.0, 2.0, 2.0]
x2= [0.0, 0.667, 1.667, 1.741, 1.981, 1.947, 2.009, 1.987, 2.006, 1.996, 2.002, 1.999, 2.001, 2.0, 2.0]
x3= [0.0, 2.0, 1.889, 2.111, 1.969, 2.037, 1.986, 2.012, 1.994, 2.004, 1.998, 2.001, 1.999, 2.0, 2.0]

x1= 2.0
x2= 2.0
x3= 2.0

after 14 iterations.

try another system of linear equations
Enter the number of equations:
```

## 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
18       Dr. Mahmoud Gamal')
19 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

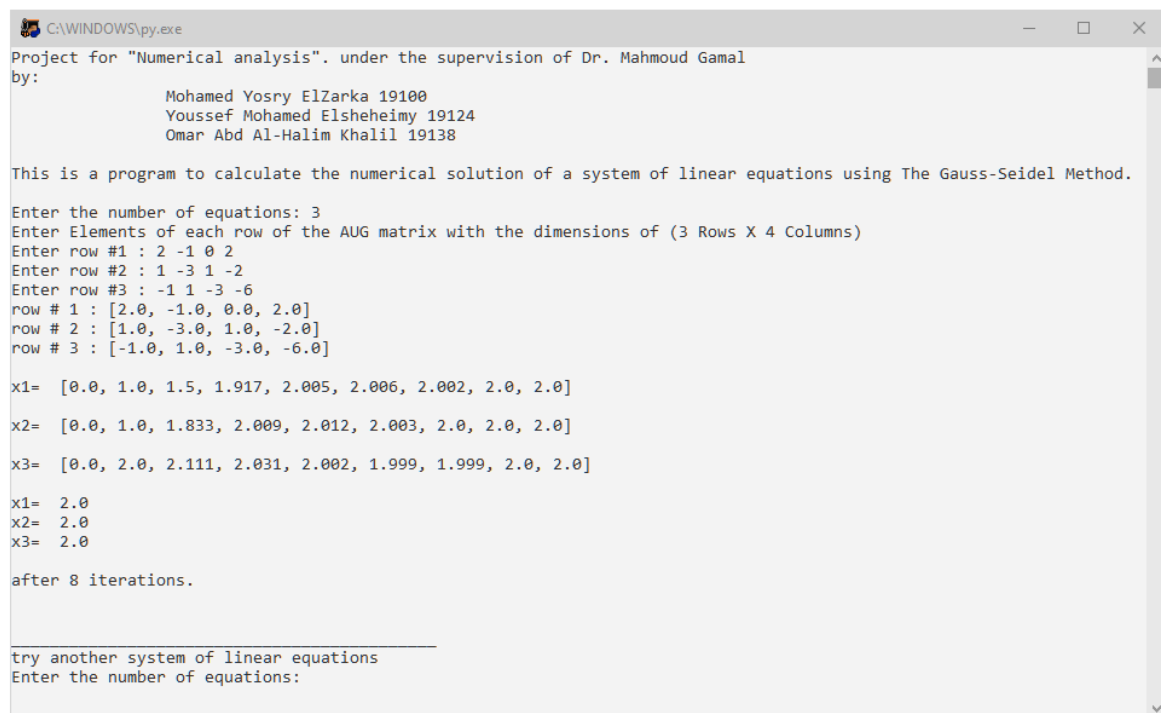
```

```

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



```

C:\WINDOWS\py.exe
Project for "Numerical analysis". under the supervision of Dr. Mahmoud Gamal
by:
    Mohamed Yosry ElZarka 19100
    Youssef Mohamed Elsheheimy 19124
    Omar Abd Al-Halim Khalil 19138

This is a program to calculate the numerical solution of a system of linear equations using The Gauss-Seidel Method.

Enter the number of equations: 3
Enter Elements of each row of the AUG matrix with the dimensions of (3 Rows X 4 Columns)
Enter row #1 : 2 -1 0 2
Enter row #2 : 1 -3 1 -2
Enter row #3 : -1 1 -3 -6
row # 1 : [2.0, -1.0, 0.0, 2.0]
row # 2 : [1.0, -3.0, 1.0, -2.0]
row # 3 : [-1.0, 1.0, -3.0, -6.0]

x1= [0.0, 1.0, 1.5, 1.917, 2.005, 2.006, 2.002, 2.0, 2.0]
x2= [0.0, 1.0, 1.833, 2.009, 2.012, 2.003, 2.0, 2.0, 2.0]
x3= [0.0, 2.0, 2.111, 2.031, 2.002, 1.999, 1.999, 2.0, 2.0]

x1= 2.0
x2= 2.0
x3= 2.0

after 8 iterations.

_____
try another system of linear equations
Enter the number of equations:

```