

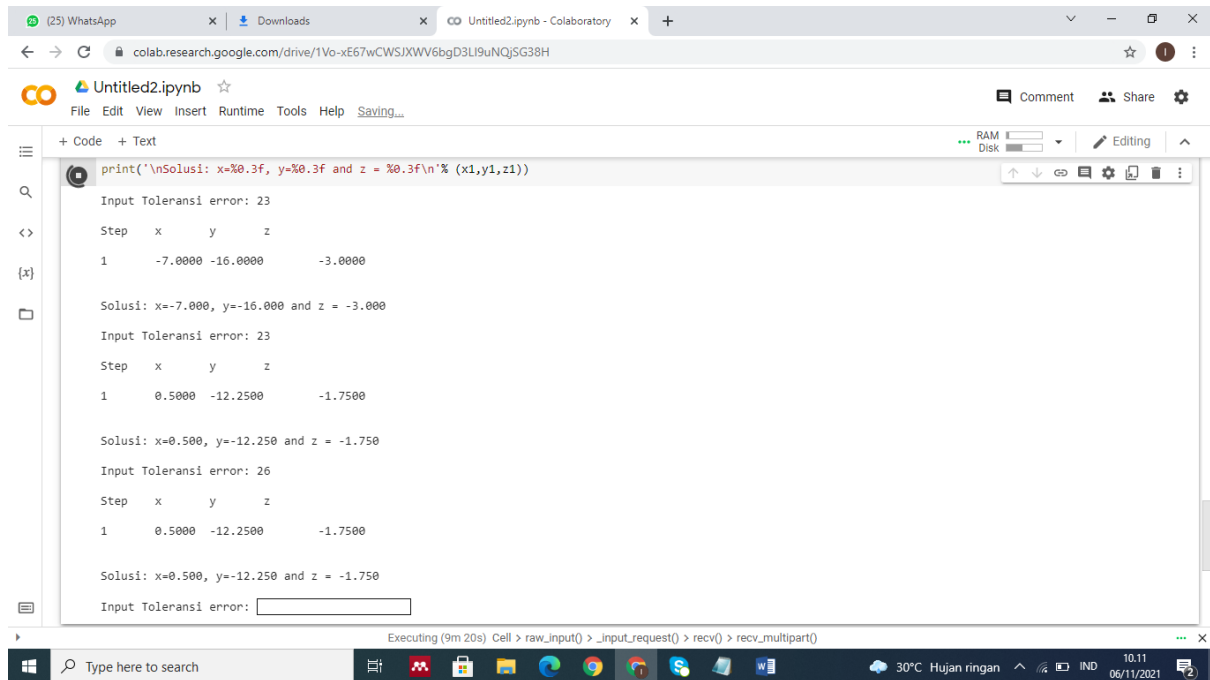
Nama : Iqbal Mahandika Putra

NPM : 202010225268

Kelas : TF3A6

PRAKTIKUM SESI 2 METODE NUMERIK

Lat Gaus Seidel



```
print('\nSolusi: x=%0.3f, y=%0.3f and z = %0.3f\n' % (x1,y1,z1))

Input Toleransi error: 23

Step  x      y      z
1     -7.0000 -16.0000 -3.0000

Solusi: x=-7.000, y=-16.000 and z = -3.000

Input Toleransi error: 23

Step  x      y      z
1     0.5000 -12.2500 -1.7500

Solusi: x=0.500, y=-12.250 and z = -1.750

Input Toleransi error: 26

Step  x      y      z
1     0.5000 -12.2500 -1.7500

Solusi: x=0.500, y=-12.250 and z = -1.750

Input Toleransi error: 
```

Iterasi Gauss Seidel

Definisikan Persamaan yang akan diselesaikan

Dalam bentuk dominan secara diagonal

Iterasi Gauss Seidel

Definisikan Persamaan yang akan diselesaikan

Dalam bentuk dominan secara diagonal

$f1 = \text{lambdax,y,z: } (-4+3*y-0*z)/4$

$f2 = \text{lambdax,y,z: } (40-2*x+5*z)/-4$

$f3 = \text{lambdax,y,z: } (14+0*x+2*y)/6$

Inisial awal

$x0 = 2$

$y0 = -8$

$z0 = 2$

```

step = 1

# Input nilai galat/error
e = float(input('Input Toleransi error: '))

# Implementasi iterasi Gauss Seidel
print('\nStep\tx\tz\n')

condition = True

while condition:
    x1 = f1(x0,y0,z0)
    y1 = f2(x1,y0,z0)
    z1 = f3(x1,y1,z0)
    print('%d\t%.4f\t%.4f\t%.4f\n' %(step, x1,y1,z1))
    e1 = abs(x0-x1);
    e2 = abs(y0-y1);
    e3 = abs(z0-z1);

    step +=1
    x0 = x1
    y0 = y1
    z0 = z1

    condition = e1>e and e2>e and e3>e
print('\nSolusi: x=%.3f, y=%.3f and z = %.3f\n' % (x1,y1,z1))

# Inisial awal
x0 = 1
y0 = 2
z0 = 2
step = 1

# Input nilai galat/error
e = float(input('Input Toleransi error: '))

# Implementasi iterasi Gauss Seidel
print('\nStep\tx\tz\n')

condition = True

while condition:
    x1 = f1(x0,y0,z0)
    y1 = f2(x1,y0,z0)
    z1 = f3(x1,y1,z0)
    print('%d\t%.4f\t%.4f\t%.4f\n' %(step, x1,y1,z1))

```

$$\begin{aligned}x_0 &= 1 \\ y_0 &= 2\end{aligned}$$

$z_0 = 2$

step = 1

Input nilai galat/error

$e = \text{float}(\text{input}(\text{'Input Toleransi error: '}))$

Implementasi iterasi Gauss Seidel

$\text{print}(\text{'\nStep\tx\ty\tz\n'})$

condition = True

while condition:

$x_1 = f_1(x_0, y_0, z_0)$

$y_1 = f_2(x_1, y_0, z_0)$

$z_1 = f_3(x_1, y_1, z_0)$

$\text{print}(\text{'%d\t%0.4f\t%0.4f\t%0.4f\n'} \% (\text{step}, x_1, y_1, z_1))$

$e_1 = \text{abs}(x_0 - x_1);$

$e_2 = \text{abs}(y_0 - y_1);$

$e_3 = \text{abs}(z_0 - z_1);$

step += 1

$x_0 = x_1$

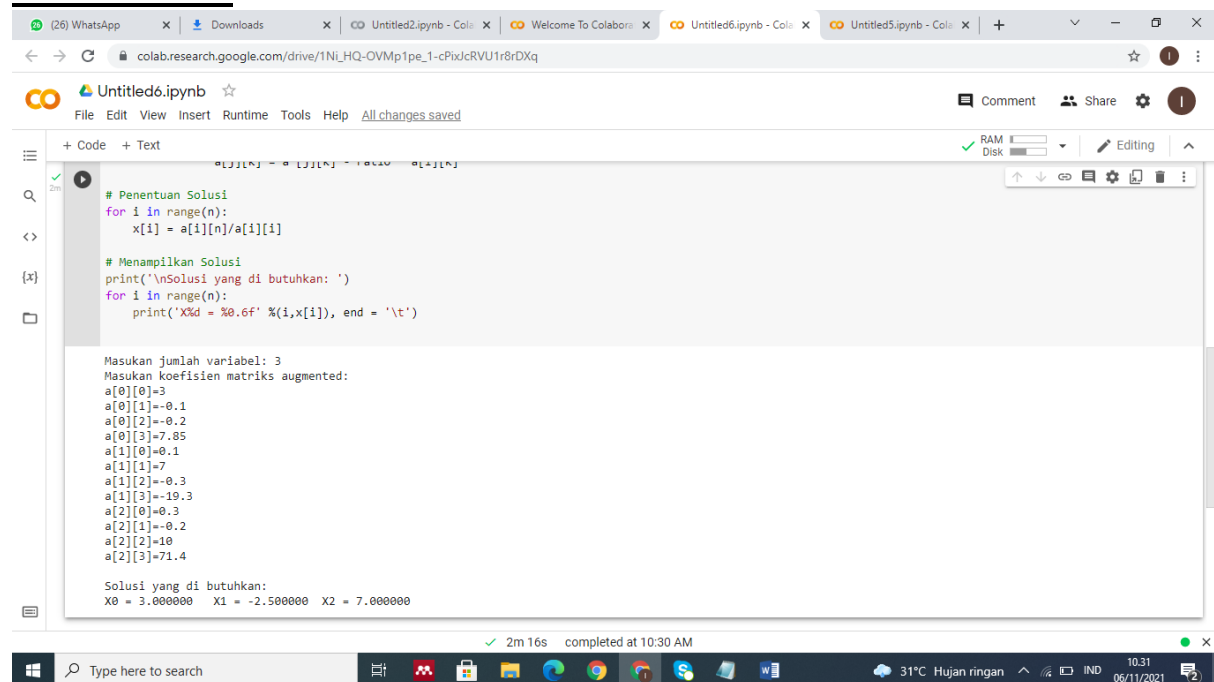
$y_0 = y_1$

$z_0 = z_1$

condition = $e_1 > e$ and $e_2 > e$ and $e_3 > e$

$\text{print}(\text{'\nSolusi: x=%0.3f, y=%0.3f and z = %0.3f\n'} \% (x_1, y_1, z_1))$

Gauss Jordan



The screenshot shows a Google Colab notebook titled 'Untitled6.ipynb'. The code in the notebook is as follows:

```
# Penentuan Solusi
for i in range(n):
    x[i] = a[i][n]/a[i][i]

# Menampilkan Solusi
print('\nSolusi yang di butuhkan: ')
for i in range(n):
    print('X%d = %0.6f' %(i,x[i]), end = '\t')
```

The output of the code is:

```
Masukan jumlah variabel: 3
Masukan koefisien matriks augmented:
a[0][0]=3
a[0][1]=-0.1
a[0][2]=-0.2
a[0][3]=7.85
a[1][0]=0.1
a[1][1]=7
a[1][2]=-0.3
a[1][3]=-19.3
a[2][0]=0.3
a[2][1]=-0.2
a[2][2]=10
a[2][3]=71.4

Solusi yang di butuhkan:
X0 = 3.000000 X1 = -2.500000 X2 = 7.000000
```

The notebook interface shows the code editor on the left and the output on the right. The bottom status bar indicates that the code was completed at 10:30 AM.

```

import numpy as np
import sys

n = int (input('Masukan jumlah variabel: '))

# Membuat array berukuran n x n+1 dan menginisiasi
# Menyimpan matriks augmented A | b
a = np.zeros((n,n+1))

# Membuat array berukuran n dan menginisiasi
# Vektor solusi
x = np.zeros(n)

# Membaca koefisien matrik augmented
print('Masukan koefisien matriks augmented: ')
for i in range(n):
    for j in range(n+1):
        a[i][j] = float(input( 'a[' +str(i)+'']['+str(j)+'']='))

# Implementasi Eliminasi Gaus Jordan
for i in range (n):
    if a[i][i] == 0.0:
        sys.exit('Divide by zero detected!: ')

    for j in range(n):
        if i != j:
            ratio = a[j][i]/a[i][i]

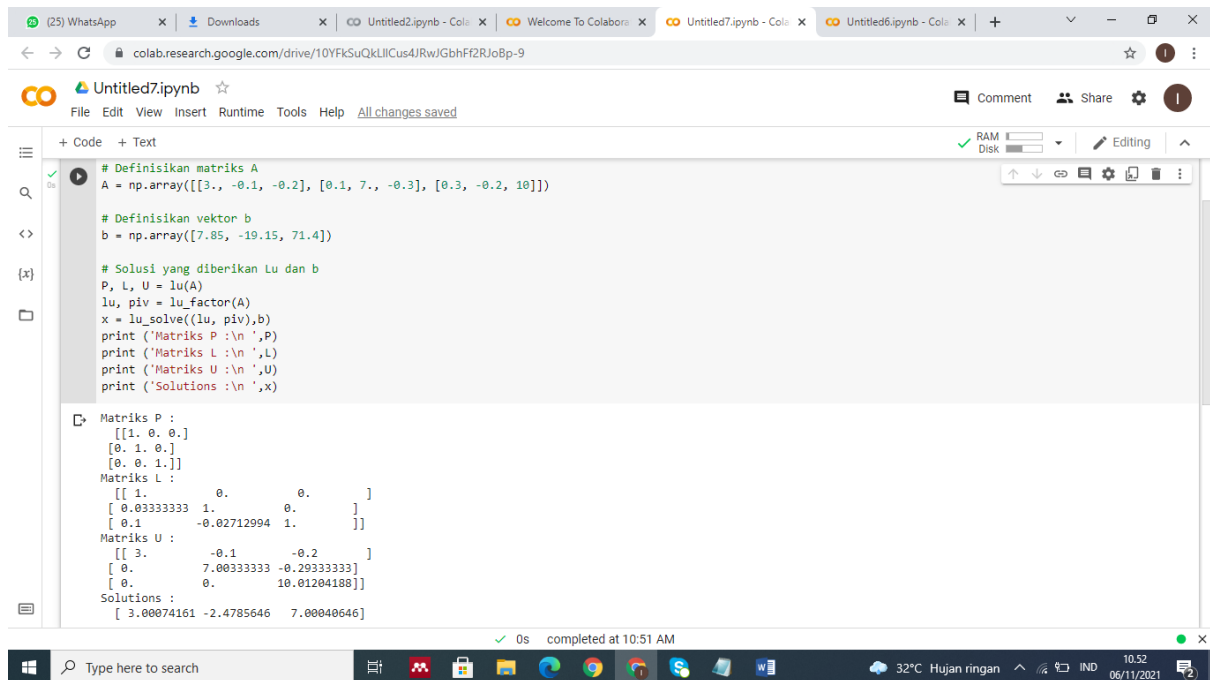
            for k in range(n+1):
                a[j][k] = a [j][k] - ratio * a[i][k]

# Penentuan Solusi
for i in range(n):
    x[i] = a[i][n]/a[i][i]

# Menampilkan Solusi
print('\nSolusi yang di butuhkan: ')
for i in range(n):
    print('X%d = %0.6f' %(i,x[i]), end = '\t')

```

Faktorisasi



The screenshot shows a Jupyter Notebook titled 'Untitled7.ipynb' in a web browser. The code defines a matrix A, a vector b, and performs LU decomposition. The output displays the matrices P, L, U, and the solution x.

```
# Definiskan matriks A
A = np.array([[3., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])

# Definiskan vektor b
b = np.array([7.85, -19.15, 71.4])

# Solusi yang diberikan Lu dan b
P, L, U = lu(A)
lu, piv = lu_factor(A)
x = lu_solve((lu, piv), b)
print ('Matriks P :\n ', P)
print ('Matriks L :\n ', L)
print ('Matriks U :\n ', U)
print ('Solutions :\n ', x)
```

Output:

```
Matriks P :
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
Matriks L :
[[1.         0.         0.        ]
 [ 0.03333333  1.         0.        ]
 [ 0.1        -0.02712994  1.        ]]
Matriks U :
[[ 3.         -0.1        -0.2        ]
 [ 0.         7.00333333 -0.29333333]
 [ 0.         0.         10.01204188]]
Solutions :
[ 3.00074161 -2.4785646  7.00040646]
```

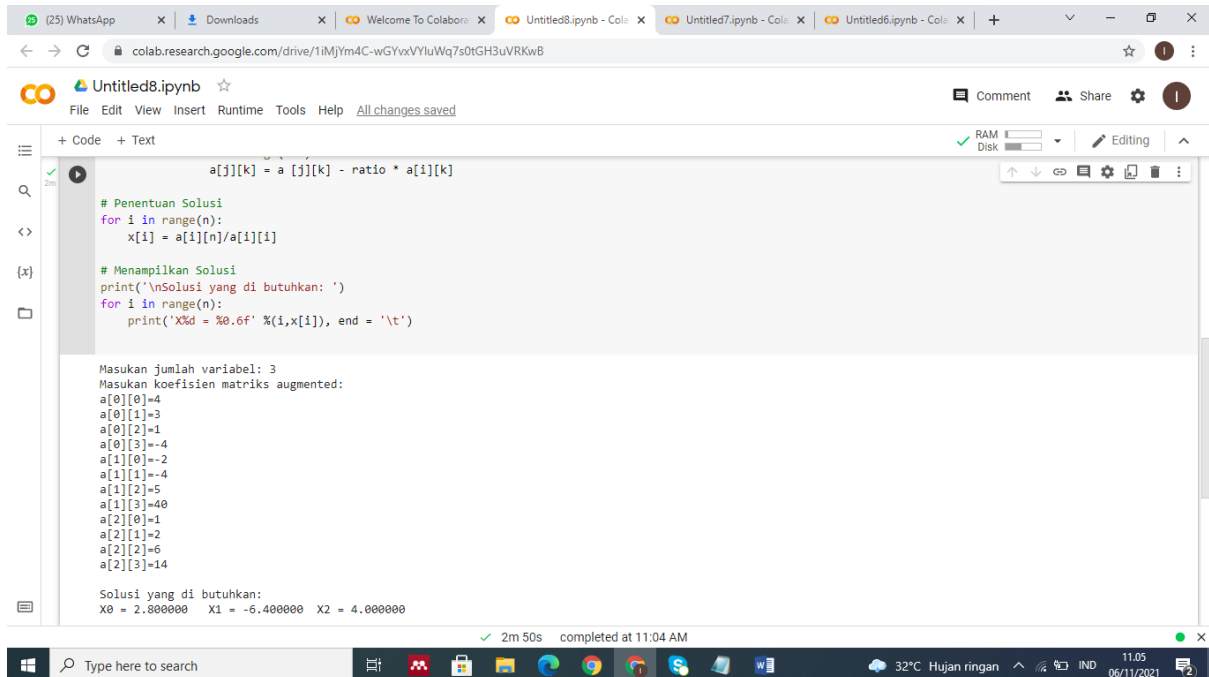
```
import scipy
from scipy.linalg import lu, lu_factor, lu_solve
import numpy as np

# Definiskan matriks A
A = np.array([[3., -0.1, -0.2], [0.1, 7., -0.3], [0.3, -0.2, 10]])

# Definiskan vektor b
b = np.array([7.85, -19.15, 71.4])

# Solusi yang diberikan Lu dan b
P, L, U = lu(A)
lu, piv = lu_factor(A)
x = lu_solve((lu, piv), b)
print ('Matriks P :\n ', P)
print ('Matriks L :\n ', L)
print ('Matriks U :\n ', U)
print ('Solutions :\n ', x)
```

Lat Gaus Jordan



```
a[j][k] = a[j][k] - ratio * a[i][k]

# Penentuan Solusi
for i in range(n):
    x[i] = a[i][n]/a[i][i]

# Menampilkan Solusi
print('\nSolusi yang di butuhkan: ')
for i in range(n):
    print('X%d = %0.6f' %(i,x[i]), end = '\t')

Masukan jumlah variabel: 3
Masukan koefisien matriks augmented:
a[0][0]=4
a[0][1]=3
a[0][2]=1
a[0][3]=-4
a[1][0]=-2
a[1][1]=-4
a[1][2]=5
a[1][3]=40
a[2][0]=1
a[2][1]=2
a[2][2]=6
a[2][3]=-14

Solusi yang di butuhkan:
X0 = 2.800000 X1 = -6.400000 X2 = 4.000000
```

```
import numpy as np
```

```
import sys
```

```
n = int (input('Masukan jumlah variabel: '))
```

```
# Membuat array berukuran n x n+1 dan menginisiasi
```

```
# Menyimpan matriks augmented A | b
```

```
a = np.zeros((n,n+1))
```

```
# Membuat array berukuran n dan menginisiasi
```

```
# Vektor solusi
```

```
x = np.zeros(n)
```

```
# Membaca koefisien matriks augmented
```

```
print('Masukan koefisien matriks augmented: ')
```

```
for i in range(n):
```

```
    for j in range(n+1):
```

```
        a[i][j] = float(input( 'a['+str(i)+']['+str(j)+']='))
```

```
# Implementasi Eliminasi Gaus Jordan
```

```
for i in range (n):
```

```
    if a[i][i] == 0.0:
```

```
        sys.exit('Divide by zero detected!: ')
```

```
    for j in range(n):
```

```

    if i != j:
        ratio = a[j][i]/a[i][i]

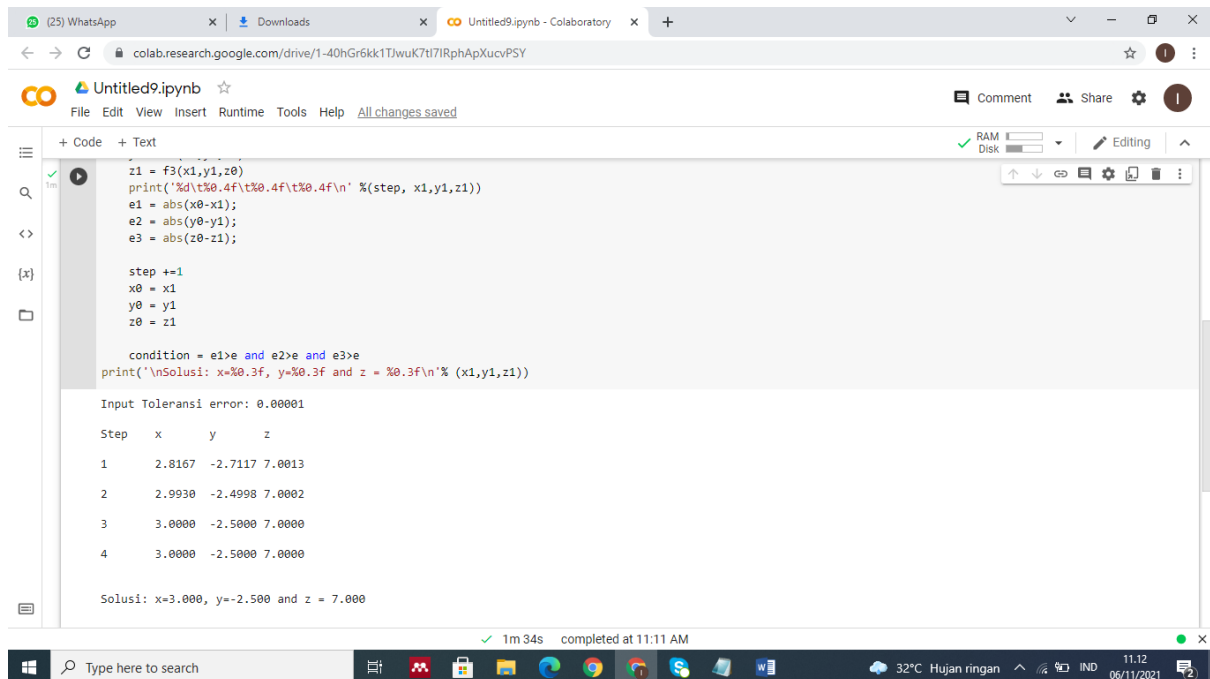
    for k in range(n+1):
        a[j][k] = a [j][k] - ratio * a[i][k]

# Penentuan Solusi
for i in range(n):
    x[i] = a[i][n]/a[i][i]

# Menampilkan Solusi
print("\nSolusi yang di butuhkan: ")
for i in range(n):
    print('X%d = %0.6f' %(i,x[i]), end = '\t')

```

Gaus Seidel



```

z1 = f3(x1,y1,z0)
print('%d\t%0.4f\t%0.4f\t%0.4f\n' %(step, x1,y1,z1))
e1 = abs(x0-x1);
e2 = abs(y0-y1);
e3 = abs(z0-z1);

step +=1
x0 = x1
y0 = y1
z0 = z1

condition = e1>e and e2>e and e3>e
print('\nSolusi: x=%0.3f, y=%0.3f and z = %0.3f\n' (x1,y1,z1))

Input Toleransi error: 0.00001

Step   x       y       z
1      2.8167  -2.7117  7.0013
2      2.9930  -2.4998  7.0002
3      3.0000  -2.5000  7.0000
4      3.0000  -2.5000  7.0000

Solusi: x=3.000, y=-2.500 and z = 7.000

```

Iterasi Gauss Seidel

Definisikan Persamaan yang akan diselesaikan

Dalam bentuk dominan secara diagonal

$f1 = \text{lambda } x,y,z: (7.85+0.1*y+0.2*z)/3$

$f2 = \text{lambda } x,y,z: (-19.3-0.1*x+0.3*z)/7$

$f3 = \text{lambda } x,y,z: (71.4-0.3*x+0.2*y)/10$

Inisial awal

$x0 = 1$

$y0 = 2$


```

z0 = 2
step = 1

# Input nilai galat/error
e = float(input('Input Toleransi error: '))

# Implementasi iterasi Gauss Seidel
print('\nStep\tx\ty\tz\n')

condition = True

while condition:
    x1 = f1(x0,y0,z0)
    y1 = f2(x1,y0,z0)
    z1 = f3(x1,y1,z0)
    print('%d\t%0.4f\t%0.4f\t%0.4f\n' %(step, x1,y1,z1))
    e1 = abs(x0-x1);
    e2 = abs(y0-y1);
    e3 = abs(z0-z1);

    step +=1
    x0 = x1
    y0 = y1
    z0 = z1

    condition = e1>e and e2>e and e3>e
print('\nSolusi: x=%0.3f, y=%0.3f and z = %0.3f\n' % (x1,y1,z1))

```