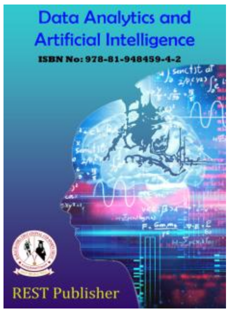
Data Analytics and Artificial Intelligence

Vol: 3(2), 2023

REST Publisher; ISBN: 978-81-948459-4-2

Website: http://restpublisher.com/book-series/daai/

DOI: https://doi.org/10.46632/daai/3/2/37



Certain Iterative Methods to Solve System of Equations by Python Programming

*** [1]ThangaMurugeshwari.V, [2]Shobanapriya.M, [3]Abhinaya. B**

St. Joseph's College of Arts and Science for Women, Hosur, Tamil Nadu, India.
*Corresponding Author Email: shobanapriya09@gmail.com

Abstract. *In the 1980s and 1990s, a field known as scientific computing or computational science began to emerge as a result of the increasing significance of using computers to carry out numerical operations in order to solve mathematical models of the world. This paper examines numerical analysis's application from a computer science view point; see[3][4][5].In this paper, Iterative methods like Gauss Jacobi and Gauss Serial were used to solve the system of simultaneous equation by using Python Programming.*

Key Words: *Numerical Analysis, Gauss Jacobi and Gauss Seidal*

1. INTRODUCTION

Numerical techniques is essentially a discipline of mathematics where issues are resolved numerically and with the use of Now include so phisticated numerical analysis software, enabling many users to undertake modeling even if they are not familiar with the underlying mathematics. Python programming is general-purpose interpreted, interactive, object oriented and high level programming language[2].

2. GAUSS JACOBI METHOD

Consider the system of equations

$$a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n=c_1$$

$$a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n=c_2$$

$$a_{n1}x_1+a_{n2}x_2+\cdots+a_{nn}x_n=c_n$$

We assume that the coefficient matrix of this system is diagonally dominant [(i.e.) the system is a diagonal system. The above equations can be written as

$$x_1=\frac{1}{a_{11}}[c_1-a_{12}x_2-a_{13}x_3-\cdots-a_{1n}x_n]\text{-(1)}$$

$$x_2=\frac{1}{a_{22}}[c_2-a_{21}x_1-a_{23}x_3-\cdots-a_{2n}x_n]\text{-(1)}$$

$$x_n=\frac{1}{a_{nn}}[c_n-a_{n1}x_1-a_{n3}x_3-\cdots-a_{n,n-1}x_{n-1}]\text{-(1)}\qquad (n)$$

We start the initial z values for the variables $x_1, x_2, x_3, \ldots x_n$ to be

$$x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, \ldots, x_n^{(0)}$$

Using this values in (1), (2), (3), … (n) respectively we get $x^{(1)}, x^{(1)}, x^{(1)}, \ldots, x^{(1)}$

Putting$x = {}^{(1)}, x=x^{(1)},\ldots,x=x^{(1)}$in

(1),(2),(3),…(n) respectively we get the next approximations$_1x^{(2)}$,$_2x^{(2)}$,$_3x^{(2)}$,…,$_nx^{(2)}$. In general if the values of x_1,$_2$,x_3,…x_n in ther^{th} alteration are$_1x^{(n)}$, $_2x^{(n)}$, $_3x^{(n)}$,..., $_nx^{(n)}$ then

$$x_1^{(r+1)} = \frac{1}{a_{11}}[c_1 - a_{12}x_2^{(r)} - a_{13}x_3^{(r)} - a_{1n}x_n^{(r)}]$$

$$x_2^{(r+1)} = \frac{1}{a_{22}}[c_2 - a_{21}x_1^{(r)} - a_{23}x_3^{(r)} - \cdots - a_{2n}x_n^{(r)}]$$

$$z_1 = \frac{1}{15}[54.8] = 3.6533$$

$$x_n^{(r+1)} = \frac{1}{a_{nn}}[c_n - a_{n1}x_1^{(r)} - a_{n2}x_2^{(r)} - a_{n,n-1}x_n^{(r)}]$$

Solving equations using Jacobi's iteration method

3x+4y+15z=54.8;
x+12y+3z=39.66;
10x+y-2z=7.74

Solution: Coefficient matrix of the given set

Of equation is $A = \begin{bmatrix} 3 & 4 & 15 \\ 1 & 12 & 3 \\ 10 & 1 & -2 \end{bmatrix}$

We note that A is not diagonally dominant

Also, $A \sim \begin{bmatrix} 10 & 1 & -2 \\ 1 & 12 & 3 \\ 3 & 4 & 15 \end{bmatrix} R_1 \leftrightarrow R_3$

Which is diagonally dominant. The given system becomes

10x+y-2z =7.74..............................(1)
x+12y+3z=39.66..................................(2)
3x+4y+15z =54.8 (3)

From (1), (2)and we gets

$X = \frac{1}{10}[7.74 - y + 2z]$.......................(4)

$Y = \frac{1}{12}[39.66 - x - 3z]$............................(5

$Z = \frac{1}{15}[54.8 - 3x - 4y]$.............................(6)

First Iteration:

Let the initial value be $x_o = y_0 = z = _0$ 0. From (4), (5) and (6)

$x_1 = \frac{1}{10}[7.74] = 0.774$

$y_1 = \frac{1}{12}[39.66] = 3.305$

Second Iteration:

$x_2 = \frac{1}{10}[7.74 - y_1 + 2z_1] = \frac{1}{10}[7.74 - 3.305 + 7.3066] = 1.1742$

$y_2 = \frac{1}{12}[39.66 - x_1 - 3z_1] = \frac{1}{12}[39.66 - 0.774 - 10.9599] = 2.3272$

$z_2 = \frac{1}{15}[54.8 - 3x_1 - 4y_1] = \frac{1}{15}[54.8 - 2.322 - 13.22] = 2.6172$

Third Iteration:

$x_3 = \frac{1}{10}[7.74 - y_2 + 2z_2] = \frac{1}{10}[7.74 - 2.3272 + 5.2344] = 1.0647$

$y_3 = \frac{1}{12}[39.66 - x_2 - 3z_2] = \frac{1}{12}[39.66 - 1.1742 - 7.8516] = 2.5529$

$z_3 = \frac{1}{15}[54.8 - 3x_2 - 4y_2] = \frac{1}{15}[54.8 - 3.5226 - 9.3088] = 2.7979$

Fourth Iteration:

$x_4 = \frac{1}{10}[7.74 - y_3 + 2z_3] = \frac{1}{10}[7.74 - 2.5529 + 5.5958] = 1.0783$

$y_4 = \frac{1}{12}[39.66 - x_3 - 3z_3] = \frac{1}{12}[39.66 - 1.0647 - 8.3937] = 2.5168$

$z_4 = \frac{1}{15}[54.8 - 3x_3 - 4y_3] = \frac{1}{15}[54.8 - 3.1941 - 10.2116] = 2.7596$

Fifth Iteration:

$x_5 = \frac{1}{10}[7.74 - y_4 + 2z_4] = \frac{1}{10}[7.74 - 2.5168 + 5.5192] = 1.0742$

$y_5 = \frac{1}{12}[39.66 - x_4 - 3z_4] = \frac{1}{12}[39.66 - 1.0783 - 8.2788] = 2.5252$

$z_5 = \frac{1}{15}[54.8 - 3x_4 - 4y_4] = \frac{1}{15}[54.8 - 3.2349 - 10.0672] = 2.7665$

Sixth Iteration:

$x_6 = \frac{1}{10}[7.74\text{-}y_5+2z_5]=\frac{1}{10}[7.74\text{-}2.5252+5.533]=1.0783$

$y_6 = \frac{1}{12}[39.66\text{-}x_5\text{-}3z_5]= \frac{1}{12}[39.66\text{-}1.0742\text{-}8.2995]=2.5239$

$z_6 = \frac{1}{15}[54.8\text{-}3x_5\text{-}4y_5]= \frac{1}{15}[54.8\text{-}3.2226\text{-}10.1008]=2.7651$

Seventh Iteration:

$x_7 = \frac{1}{10}[7.74\text{-}y_6+2z_6]= \frac{1}{10}[7.74\text{-}2.5239+5.5302]=1.0746$

$y_7 = \frac{1}{12}[39.66\text{-}x_6\text{-}3z_6]=\frac{1}{12}[39.66\text{-}1.0748\text{-}8.2953]=2.5242$

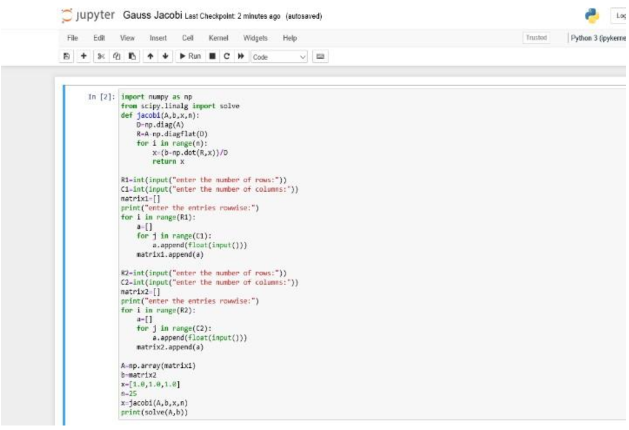
$z_7 = \frac{1}{15}[54.8\text{-}3x_6\text{-}4y_6]=\frac{1}{15}[54.8\text{-}3.2244\text{-}10.0956]=2.7653$

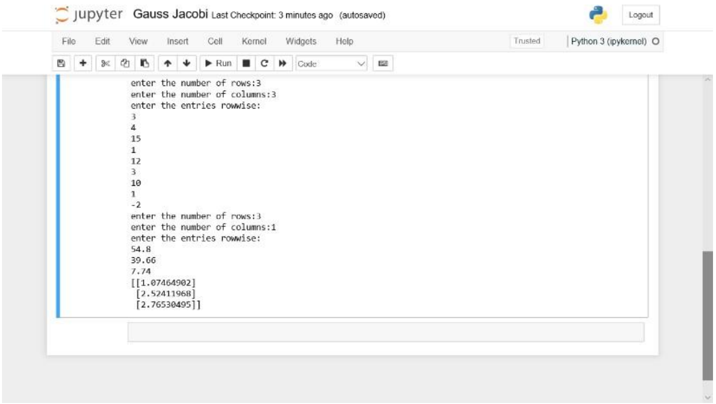
After 7 iteration the difference in 6^{th} and 7^{th} iteration are negligible. Hence the solution of the system is given by x=1.075; y =2.524; and z=2.765

Python Code to Solve Simultaneous Equation by Gauss Jacobi:

```
import numpy as np from scipy. linalg import solve defjacobi (A,b,x,n):
D=np. diag(A)
R=A-np. diag flat(D)
For iin range (n):
x= (b-np.dot(R,x))/D return x
R1=int(input("enter the number of rows:"))
C1=int(input("enter the number of columns:"))

matrix1= []
Print ("enter the entries row wise :")
For i in range (R1):a=[]
For j in range (C1): append (float (input ())) matrix1.append (a)
R2=int (input("enter the number of rows:"))
C2=int (input("enter the number of columns:"))

matrix2=[]
print("enter the entries row wise:")for i in range(R2): a=[] for j in range (C2):a. append (float (input()))
matrix2. Append (a)
A= np. Array (matrix1)
b= matrix2
x= [1.0,1.0,1.0]
n=25
x=Jacobi (A,b,x,n)
print (solve(A,b))
```

Illustration of Gauss Jacobi in Jupyter Note





3. GAUSS SEIDAL METHOD

Gauss-Seidelite ration method is finement of Gauss-Jacobi method. As in Jacobi method let

$$x_1 = \frac{1}{a_{11}}[c_1 - a_{12}\, x_2 - a_{13}\, x_3 - \cdots - a_{1n}\, x_n] \quad \ldots\ldots\ldots\ldots\ldots(1)$$

$$x_2 = \frac{1}{a_{22}}[c_2 - a_{21}\, x_1 - a_{23}\, x_3 - \cdots - a_{2n}x_n] \quad \text{---------} \quad (2)$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$x_n = \frac{1}{a_{nn}}[c_n - a_{n1}x_2 - a_{n2}x_3 - \cdots - a_{n-1}x_n] \text{---------}(n)$$

We start with the initial values $^{(0)},^{(0)},\ldots,x^{(0)}$ and we get from(1)

$$x_1^{(1)} = \frac{1}{a_{11}}[c_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)} - \dots - a_{1n}x_n^{(0)}]$$

In the second equation we use $x_1^{(1)}$ for x_1 And $x_3^{(0)}$ for x_3 etc. and $x_n^{(0)}$ for x_n. [In the Jacobi Method we use $x_1^{(0)}$ for x_1].Thus we get,

$$x_2^{(1)} = \frac{1}{a_{22}}[c_2 - a_{11}x_1^{(1)} - a_{13}x_3^{(0)} - \dots - a_{2n}x_n^{(0)}]$$

Proceeding like this we find the first iteration values as $x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \dots, x_n^{(1)}$ In general if the values of the variables in the r^th iteration are given by

$$x_1^{(r+1)} = \frac{1}{a_{11}}[c_1 - a_{12}x_2^{(r)} - a_{13}x_3^{(r)} - \dots - a_{1n}x_n^{(r)}]$$

$$x_2^{(r+1)} = \frac{1}{a_{22}}[c_2 - a_{11}x_1^{(r)} - a_{13}x_3^{(r)} - \dots - a_{2n}x_n^{(r)}]$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$x_n^{(r+1)} = \frac{1}{a_{nn}}[c_n - a_{n1}x_1^{(r+1)} - a_{n2}x_2^{(r+1)} - \dots - a_{n,n-1}x_{n-1}^{(r+1)}]$$

Solving system of equation using Gauss Seidel iteration method:

6x+15y+2z=72;

x+y+54z=110;

27x+6y-z=85

Solution: Coefficient matrix of the given system of equation is

$$A=\begin{bmatrix} 6 & 15 & 2 \\ 1 & 1 & 54 \\ 27 & 6 & -1 \end{bmatrix}$$ We note that A is not

Diagonally dominant. However, it can be made diagonally dominant the rows

R_1->R_2andthenR_2<->R_3

$$A=\begin{bmatrix} 27 & 6 & -1 \\ 6 & 15 & 2 \\ 1 & 1 & 54 \end{bmatrix}$$

Hence the corresponding system of equation is

27x+6y-z=85

6x+15y+2z =72

x+y+54z =110

The above system of equation can be rewritten as

$X=\frac{1}{27}(85-6y+z)$ .... (1)

$Y=\frac{1}{15}(72-6x-2z)$ .... (2)

$Z=\frac{1}{15}(110-x-y)$ .... (3)

First Iteration:

Putting y=0 and z=0in (1) we get $x=\frac{85}{27}=$ 3.1481,

Putting x=3.1481 and z=0 in (2) we get $y=\frac{1}{15}[72-6\times 3.1481]=3.5408$

Putting x= 3.1481, y=3.5408 in (3)

We get $z=\frac{1}{54}[110-3.1481-3.5408]=1.9132$

Second Iteration:

Putting y= 3.5408 and z=1.9132 in (1) we get $x=\frac{1}{27}[85-6\times3.5408+1.9132]=$ 2.4322

Putting x=2.4322 and z= 1.9132 in (2) we get $y=\frac{1}{15}[72-6\times2.4322-2\times1.9132]=$ 3.572

Putting x=2.4322 and y=3.572 in (3)

We get $z=\frac{1}{54}[110-2.4322-3.572]=1.9258$

Third Iteration:

Putting y= 3.572 and z=1.9258 in (1) we get $x=\frac{1}{27}[85-6\times3.572+1.9258]=2.4257$

Putting x=2.4257 and z=1.9258in (2) we get $y=\frac{1}{15}[72-6\times2.4257-2\times1.9258]=$ 3.5729

Putting x=2.4257and y=3.5729in (3) we get $z=\frac{1}{54}[110-2.4257-3.5729]=$ 1.926

Fourth Iteration:

Putting y= 3.5729and z= 1.926 in (1)we get $x=\frac{1}{27}[85-6\times3.5729+1.926]=$ 2.4255

Putting x=2.4255 and z=1.926 in (2)

we get $y=\frac{1}{15}$ [72-6×2.4255-2×1.926]= 3.573

Putting x=2.4255 and y=3.573 in (3)

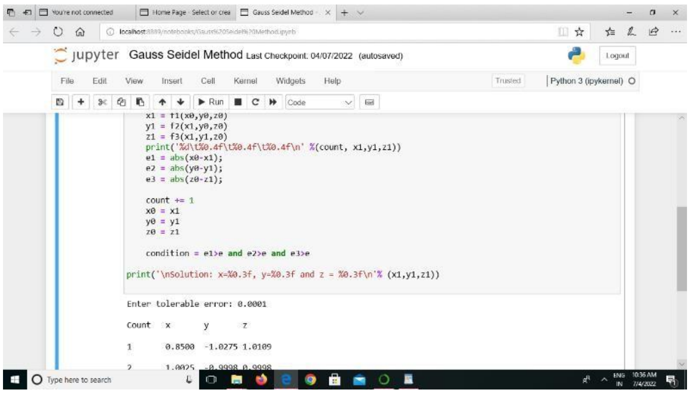
We get $\frac{1}{54}$ [110-2.4255-3.573]=1.926

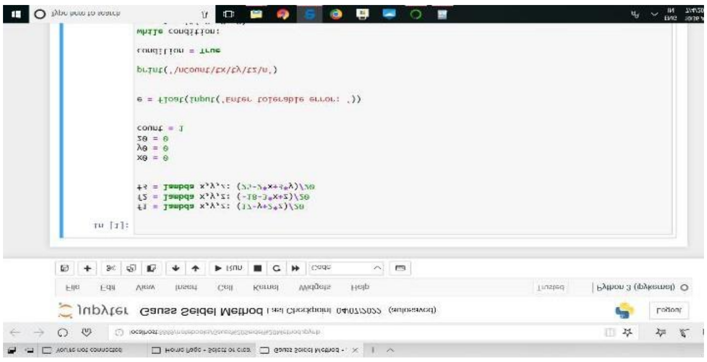
The values of x, y, z in the third and fourth iteration is almost equal. The values of x, y, z in the third and fourth iteration is almost equal. Hence the roots of the system are x=2.4255,y=3.573, z=1.926

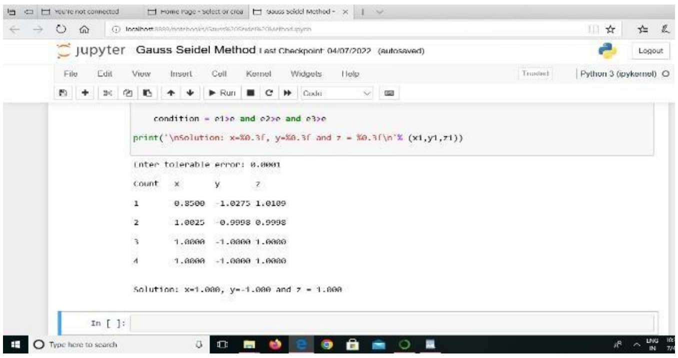
Python Code to Solve Simultaneous Equation by Gauss Seidal:

```
f1 = lambda x, y, z: (85-6*y + z)/27f2 = lambda x, y, z: (72-6*x- 2*z)/15f3=lambda x, y, z:(110-x- y)/54
x0 =0, y0 =0, z0=0, count= 1

e = float (input ('Enter tolerable error:')) print('\n Count\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' %(count,x1,y1,z1))
                    e1=abs(x0-x1);
                    e2=abs(y0- y1);
                    e3=abs(z0-z1);
count+=1, x0 =x1, y0 =y1, z0=z, condition=e1>e ande2>eande3>e
print('\n Solution : x=%0.3f,y=%0.3f and z= %0.3f\n'%(x1,y1,z1))
```

Illustration of Gauss Seidal in Jupyter Note:







4. CONCLUSION

In this paper, simultaneous equations are solved by Gauss Jacobi and Gauss Seidal methods. Our future works includes exploring various iterative methods to solve system of simultaneous equation by python Programming.

REFERENCES

[1].S.Arumugam, Ahangapandi Issac, Somasundaram.A, ”Numericalmethods”, citech publications.
[2].DhinaSuresh, Aswini.G, Jayanthi.P, AnushaPrem, ”Problem Solving using Python”, Charulatha Publications.
[3].L.Fosdick, E.Jessup, C.Schauble, and G. Domik (1996) An Introduction to High-Performance Scientific Computing, MIT Press.
[4].H. Goldstine (1977) A History of Numerical Analysis: From the 16th Throughthe19thCentury,Springer-Verlag.
[5].G. GolubandC.Van M. Heath (1997) Scientific Computing: An Introductory Survey, McGraw-Hill Inc.