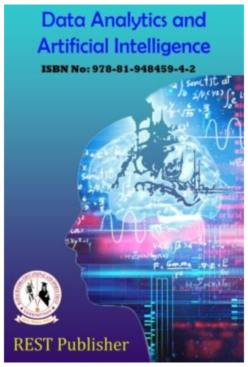
Data Analytics and Artificial Intelligence

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/3/2/25


Data Analytics and Artificial Intelligence
ISBN No: 978-81-948459-4-2
REST Publisher

Python Programming to Solve Linear Equations by Gaussian methods

***Shobana priya. M, Thanga Murugeshwari. V, Rezvana Parveen. S**

St. Joseph's College of Arts and Science for Women , Hosur, Tamil Nadu, India.

**Corresponding Author E-mail: shobanapriya09@gmail.com*

Abstract. *The development to f numerical processes was greatly aided by Newton and Leibniz's invention of calculus, which produced precise mathematical representations of physical reality first and then in other disciplines such as engineering, healthcare, and commerce. Typically, these mathematical models cannot be solved explicitly; therefore, approximate solutions must be obtained using numerical approaches [3][4][5].In this paper, Python Programming is used to solve system of linear equations by Gauss Elimination and Gauss Jordan method.*

Keywords: *Numerical Analysis, Gauss Elimination and Gauss Jordan method.*

1. INTRODUCTION

In applied mathematics, many issues require solving system so f linear equations, where in the linear system occasionally occurs naturally and in other situations as a step in the solution process. Typically, linear systems are stated as Ax=bi where A is the system's coefficient matrix, x is a column vector of unknownvariables(x1... xn), and b is as pacific column vector. Most of the time, it is now thought to be quite simple to solve linear systems with up to n = 1000 variables. For linear systems of small to inter mediate size, Gaussian elimination and its variations are preferred in numerical approach. Python supports functional and structured Programming methods as well as OOP. It Can be used as scripting language [2].

2. GAUSS ELIMINATION METHOD

Gauss elimination method is a direct method which consists of transforming the given system of simultaneous equation to an equivalent *upper triangular system*. From this transformed system the required solution can be obtained by the method of back substitution. Consider, the system of *n* equations in *n* unknowns given by A X = B where *A* is the coefficient matrix.

The augmented matrix is

$$a_{11}\ a_{12}\quad \cdots a_n$$

$$(A, B)=(a21a22\ \ldots\quad a2n|b2)$$

$$\vdots\quad \vdots\ \ldots\quad \vdots\quad \vdots$$

$$an1\ an2\qquad ann\quad b_n$$

To transform the system to an equivalent upper triangular system, we use the following row operations.

The row operation $R_i \to R_i -$

$\underline{a_{i1}}R$; =2,3,....,$n$makes all

$a11$

The entries a_{21}, a_{31}... a_{n1}in the first column zeros. Here the first equation is the *pivotal equation*
a≠0 is called *pivot* and$-\underline{ai1}$
$a11$

for$i$=2, 3… are called *Multipliers* for first elimination. If a_{11}=0, we inter change the first Row with another suitable row so as to have a_{11}≠0.
Next we do the row operation, $R_i \rightarrow R_i - \underline{a2i}$; i=3,4,….,n.
$a22$
This makes all entries a_{32}, a_{42}... a_{n2}inthesecondcolumnzero.
In general the row operation$R_i \rightarrow R_{i-1} - \underline{aik}R_k$; =$k$+1, k+ 2,…, will make all entries
$a_{k+1,,+2,k}$,...,a_{nk} in the k^{th} Column zero.

Hence the given system of equations is reduced to the form $UX= B$where $U$is an upper triangular matrix. The required solution can be obtained by the method of back substitution

Solving the system of equation using Gaussian elimination method

x + y + z = 9
2x–3y+4z=13
3x +4y +5z=40

Solution: The given set of equations can be written as

$$\begin{bmatrix} 1 & 1 & 1 \\ 4 & -3 & 4 \\ 3 & 4 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 9 \\ 13 \\ 40 \end{bmatrix}$$

The augmented matrix is

$$(A,B) = \left(\begin{array}{ccc|c} 1 & 1 & 1 & 9 \\ 4 & -3 & 4 & 13 \\ 3 & 4 & 5 & 40 \end{array}\right)$$

$$(A,B) \sim \left(\begin{array}{ccc|c} 1 & 1 & 1 & 9 \\ 0 & -5 & 2 & -5 \\ 3 & 4 & 5 & 40 \end{array}\right) R_2 \rightarrow R_2 - 3\,2R_1$$

$$\sim \left(\begin{array}{ccc|c} 1 & 1 & 1 & 9 \\ 0 & -5 & 2 & -5 \\ 0 & 1 & 2 & 13 \end{array}\right) R_3 \rightarrow R_3 - 3R_1$$

$$\sim \left(\begin{array}{ccc|c} 1 & 1 & 1 & 9 \\ 0 & -5 & 2 & -5 \\ 0 & 0 & \frac{12}{5} & 12 \end{array}\right) R_3 \rightarrow R_3 - 5R_1$$

∴ The given system of equation reduces to the system

$$\frac{12}{5} z = 12$$
-5y+2z =-5
x +y +z=9

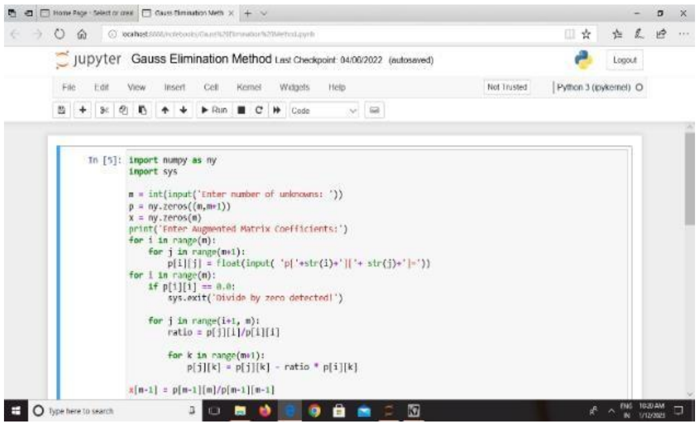
Now, by back substitution values we obtain the solution

∴x =1;y = 3;z=5.

Python Code to Solve Simultaneous Equation by Gauss Elimination

```
Import numpy as ny import sys
m = int(input('Enter number of unknowns:'))
p = ny.zeros((m,m+1))x=ny.zeros(m)
Print ('Enter Augmented Matrix
Coefficients :')
Foriin range (m):
Forjin range (m+1):
p[i][j] = float(input( 'p['+str(i)+']['+str(j)+']='))
for i in range(m):ifp[i][i]==0.0:
sys.exit('Divide by zero detected!')forj in range(i+1, m):
ratio = p[j][i]/p[i][i]fork in range(m+1):
p[j][k]=p[j][k]- ratio *p[i][k]
x[m-1]=p[m-1][m]/p[m-1][m-1]
For iin range(m-2,-1,-1):
x[i]=p[i][m]
For jin range(i+1,m):x[i]=x[i]-p[i][j]*x[j]
x[i]=x[i]/p[i][i]
Print ('\nThe solution is: ')fori in range(m):
Print ('X%d=%0.2f'%(i,x[i]),end='\t')
```

Illustration of Gauss Elimination in Jupyter Note



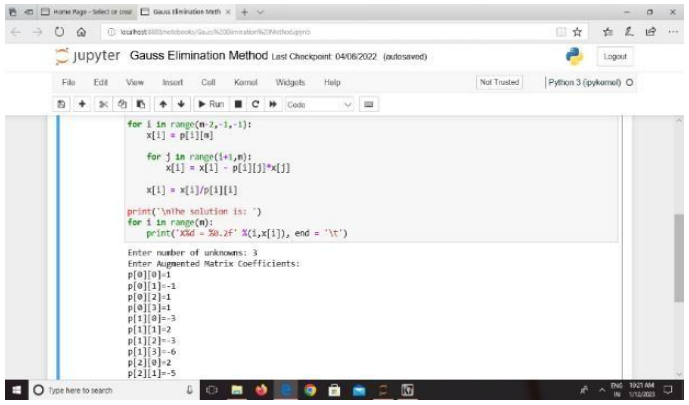
```
In [5]: import numpy as ny
        import sys

        m = int(input('Enter number of unknowns: '))
        p = ny.zeros((m,m+1))
        x = ny.zeros(m)
        print('Enter Augmented Matrix Coefficients:')
        for i in range(m):
            for j in range(m+1):
                p[i][j] = float(input( 'p['+str(i)+']['+ str(j)+']='))
        for i in range(m):
            if p[i][i] == 0.0:
                sys.exit('Divide by zero detected!')

            for j in range(i+1, m):
                ratio = p[j][i]/p[i][i]

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

        x[m-1] = p[m-1][m]/p[m-1][m-1]
```



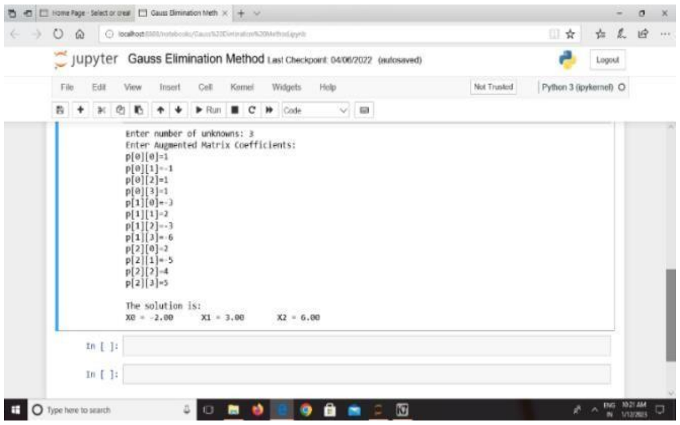
```
        for i in range(m-2,-1,-1):
            x[i] = p[i][m]

            for j in range(i+1,m):
                x[i] = x[i] - p[i][j]*x[j]

            x[i] = x[i]/p[i][i]

        print('\nThe solution is: ')
        for i in range(m):
            print('X%d = %0.2f' %(i,x[i]), end = '\t')

        Enter number of unknowns: 3
        Enter Augmented Matrix Coefficients:
        p[0][0]=1
        p[0][1]=-1
        p[0][2]=1
        p[0][3]=1
        p[1][0]=-3
        p[1][1]=2
        p[1][2]=-3
        p[1][3]=-6
        p[2][0]=2
        p[2][1]=-5
```


jupyter Gauss Elimination Method Last Checkpoint: 04/06/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
Enter number of unknowns: 3
Enter Augmented Matrix Coefficients:
p[0][0]=1
p[0][1]=-1
p[0][2]=1
p[0][3]=1
p[1][0]=-3
p[1][1]=2
p[1][2]=-3
p[1][3]=-6
p[2][0]=2
p[2][1]=-5
p[2][2]=4
p[2][3]=5

The solution is:
X0 = -2.00      X1 = 3.00       X2 = 6.00
```

3. GAUSS JORDAN METHOD

Consider the system of equation A X = B. If A is a diagonal matrix the given system reduces to

a_{11} 0 0 ⋯ 0 x_1 x_1
[0 a_{22}0 … 0[x2]=[x2]
⋮ ⋮ ⋯ ⋮ ⋮ ⋮
0 0 a_n] x_n x_n

This system reduces to the following n Equations.

a11x1=b1;22x2= b_2; … … … … … . . ;x_n= b_n.Hence we get the solution directly as

x_1= 1
——; 2=
a11 b_2
——;………;x_n
a22
b_n
=
ann

The method of obtaining the solution of the system of equation by reducing the matrix A to a diagonal matrix is known as Gauss Jordan elimination method.

Solving the system of equation using Gauss Jordan method.

5x - 2y + 3z = 18, x – 7y - 3z = -22, 2x -y +6z=22

Solution: The augmented matrix

The given set of equations can be written as

5 −2 3 18
(A, B)= (1 7 −3|−22)
2 −1 6 22

1 7 −3−22
~ (5 −2 3|18)$_1$↔R_2
2 −1 6 22

1 7 −3−22
~(0 −37 18|128)R_2→R_2−5R_1
2 −1 6 22

1 7 −3−22
~(0 −37 18|128)R_3→R_3−2R_1
0 −15 1266
1 0 0 1

—~(0 1 0|−2)$R_2 \rightarrow R_2+37R_3$
0 0 13

*Th*e given system of equation reduces to the system

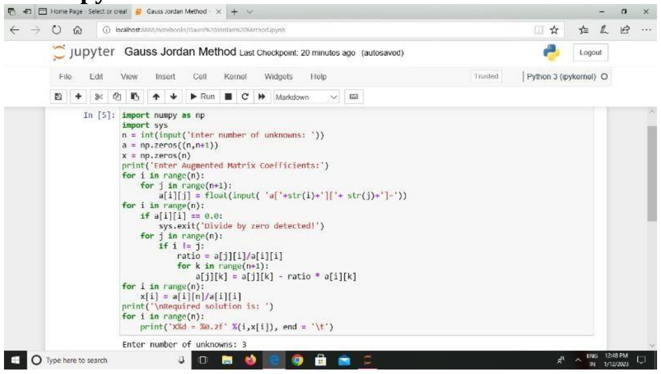
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}=\begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix}$$

∴x =1; y =-2; z = 3

Python Code to Solve Simultaneous Equation by Gauss Jordan

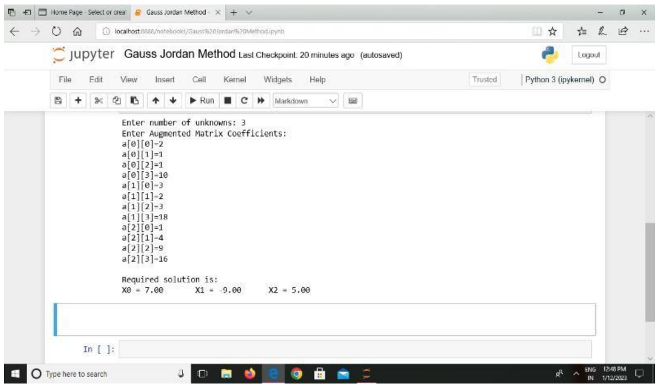
import numpy as npimport sysn=int(input('Enter number of Unknowns :'))
a = np.zero s((n,n+1))x=np.zeros(n)
Print ('Enter Augmented Matrix Coefficients :')for iin range(n):
For jin range(n+1):
a[i][j]=float(input('a['+str(i)+']['+ str(j)+']='))
For iin range(n):
ifa[i][i] ==0.0:
sys.exit('Divide by zero detected!')
for j in range(n):if i
!=j:
ratio=a[j][i]/a[i][i]
fork in range(n+1):
a[j][k] = a[j][k] - ratio * a[i][k]fori in range (n):
x[i]=a[i][n]/a[i][i]
print ('\n Required solution is: ')fori in range (n):
print ('X%d=%0.2f'%(i,x[i]), end='\t')

Illustration of Gauss Jordan in Jupyter Note



```
In [5]: import numpy as np
        import sys
        n = int(input('Enter number of unknowns: '))
        a = np.zeros((n,n+1))
        x = np.zeros(n)
        print('Enter Augmented Matrix Coefficients:')
        for i in range(n):
            for j in range(n+1):
                a[i][j] = float(input( 'a['+str(i)+']['+ str(j)+']='))
        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]
        for i in range(n):
            x[i] = a[i][n]/a[i][i]
        print('\nRequired solution is: ')
        for i in range(n):
            print('X%d = %0.2f' %(i,x[i]), end = '\t')

        Enter number of unknowns: 3
```



```
Enter number of unknowns: 3
Enter Augmented Matrix Coefficients:
a[0][0]=2
a[0][1]=1
a[0][2]=1
a[0][3]=10
a[1][0]=3
a[1][1]=2
a[1][2]=3
a[1][3]=18
a[2][0]=1
a[2][1]=4
a[2][2]=9
a[2][3]=16

Required solution is:
X0 = 7.00       X1 = -9.00      X2 = 5.00

In [ ]:
```

4. CONCLUSION

In this paper, simultaneous equations are solved by Gauss Elimination and Gauss Jordan methods. Our future works includes exploring variousiterativemethodstosolvesystemofsimultaneousequationbypythonProgramming.

REFERENCES

[1]. S. Arumugam, A hangapandiIssac, Somasundaram. A, ”Numerical methods”, Sci tech publications.
[2]. Dhina Suresh, Aswini .G,Jayanthi. P,AnushaPrem, ”Problem Solvingusing Python”, Charulatha Publications.
[3]. L.Fosdick,E.Jessup,C.Schauble,andG. Domik (1996) An Introduction to High-Performance ScientificComputing,MITPress.
[4]. H.Goldstine(1977)AHistoryofNumericalAnalysis:Fromthe16thThroughthe19thCentury,Springer-
[5]. Verlag.[18]G.GolubandC.Van M. Heath (1997) Scientific Computing:AnIntroductorySurvey,McGraw-HillInc.