Practice – Week 04

Practice – 01

1. Use Gauss Elimination to solve the following equations!

$$3x_1 - x_2 + 4x_3 = 2$$
$$17x_1 + 2x_2 + x_3 = 14$$
$$x_1 + 12x_2 - 7x_3 = 54$$

Practice – 01 – Answer

$$\begin{array}{rrr|r} 3 & -1 & 4 & 2 \\ 17 & 2 & 1 & 14 \\ 1 & 12 & -7 & 54 \end{array} \xrightarrow[\text{R3 – R1/3}]{\text{R2 – 17*R1/3}} \begin{array}{rrr|r} 3 & -1 & 4 & 2 \\ 0 & 7.677 & -21.667 & 2.667 \\ 0 & 12.333 & -8.333 & 53.333 \end{array} \xrightarrow{\text{R3 – 37*R2/23}}$$

$$\begin{array}{rrr|r} 3 & -1 & 4 & 2 \\ 0 & 7.677 & -21.667 & 2.667 \\ 0 & 0 & 26.522 & 49.043 \end{array} \longrightarrow \begin{array}{lcl} x_1 & = & 0.05902 \\ x_2 & = & 5.57377 \\ x_3 & = & 1.84918 \end{array}$$

Practice – 02

2. Use Gauss-Jordan Elimination to solve the above equations!

Practice – 02 – Answer

3	-1	4	2
17	2	1	14
1	12	-7	54

R1/3 →

1	-0.333	1.333	0.667
17	2	1	14
1	12	-7	54

R2 – 17*R1
R3 – R1 →

1	-0.333	1.333	0.667
0	7.667	-21.667	2.667
0	12.333	-8.333	53.333

3*R2/23 →

1	-0.333	1.333	0.667
0	1	-2.826	0.348
0	12.333	-8.333	53.333

R1 – R2/(-3)
R3 – 37*R2/3 →

1	0	0.391	0.783
0	1	-2.826	0.348
0	0	26.522	49.043

1830*R3/69 →

1	0	0.391	0.783
0	1	-2.826	0.348
0	0	1	1.849

R1 – 9*R3/23
R2 – (-65)*R3/23 →

Practice – 02 – Answer

$$\left.\begin{array}{ccc|c} 1 & 0 & 0 & 0.059 \\ 0 & 1 & 0 & 5.574 \\ 0 & 0 & 1 & 1.849 \end{array}\right. \Rightarrow \left.\begin{array}{ccc} x_1 & = & 0.05902 \\ x_2 & = & 5.57377 \\ x_3 & = & 1.84918 \end{array}\right|$$

Practice – 02

3. Determine the lower triangular matrix L and upper triangular matrix U from the equations!

Practice – 03 – Answer

$$\left[\begin{array}{ccc|c} 3 & -1 & 4 & 2 \\ 17 & 2 & 1 & 14 \\ 1 & 12 & -7 & 54 \end{array}\right]$$

R2 – R1*17/3
R3 – R1*1/3

$$\left[\begin{array}{ccc|c} 3 & -1 & 4 & 2 \\ 0 & 7.677 & -21.667 & 2.667 \\ 0 & 12.333 & -8.333 & 53.333 \end{array}\right]$$

R3 – R2*37/23

$$\left[\begin{array}{ccc|c} 3 & -1 & 4 & 2 \\ 0 & 7.677 & -21.667 & 2.667 \\ 0 & 0 & 26.522 & 49.043 \end{array}\right]$$

$$\text{Lower} = \begin{bmatrix} 1 & 0 & 0 \\ 17/3 & 1 & 0 \\ 1/3 & 37/23 & 1 \end{bmatrix}$$

$$\text{Upper} = \begin{bmatrix} 3 & -1 & 4 \\ 0 & 7.677 & -21.667 \\ 0 & 0 & 26.522 \end{bmatrix}$$

Python Code – 1 : Linear Eqs. Solution

In [1]:
```
import numpy as np

A = np.array([[3,-1,4],
              [17,2,1],
              [1,12,-7]])

y = np.array([2,14,54])

x1, x2, x3 = np.linalg.solve(A, y)

print(f'x1 = {x1:.5f}')
print(f'x2 = {x2:.5f}')
print(f'x3 = {x3:.5f}')
```

```
x1 = 0.05902
x2 = 5.57377
x3 = 1.84918
```

Python Code – 2 : LU Decomposition (A)

In [2]:

```
import numpy as np
import scipy.linalg as sp_la

A = np.array([[3,-1,4],
              [17,2,1],
              [1,12,-7]])

P, L, U = sp_la.lu(A)

print(P)
print(U)
print(L)
print(np.dot(L,U))
```

```
[[0. 0. 1.]
 [1. 0. 0.]
 [0. 1. 0.]]
[[17.          2.          1.        ]
 [ 0.         11.88235294 -7.05882353]
 [ 0.          0.          3.01980198]]
[[ 1.          0.          0.        ]
 [ 0.05882353  1.          0.        ]
 [ 0.17647059 -0.11386139  1.        ]]
[[17.  2.  1.]
 [ 1. 12. -7.]
 [ 3. -1.  4.]]
```

Python Code – 2 : LU Decomposition (B)

In [3]:

```
import numpy as np
import scipy.linalg as sp_la

B = np.array([[17,2,1],      # abs(17) >= abs(2) + abs(1)	--> Array with big size :
              [1,12,-7],     # abs(12) >= abs(1) + abs(-7)	--> Use Gauss-Seidel
              [3,-1,4]])     # abs(4) >= abs(3) + abs(-1)	--> for safety

P, L, U = sp_la.lu(B)

print(P)
print(U)
print(L)
print(np.dot(L,U))
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
[[17.          2.          1.        ]
 [ 0.         11.88235294 -7.05882353]
 [ 0.          0.          3.01980198]]
[[ 1.          0.          0.        ]
 [ 0.05882353  1.          0.        ]
 [ 0.17647059 -0.11386139  1.        ]]
[[17.  2.  1.]
 [ 1. 12. -7.]
 [ 3. -1.  4.]]
```

Python Code – 2 : LU Decomposition – **P**

This **permutation matrix** record how do we change the order of the equations for easier calculation purposes (for example, if first element in first row is zero, it can not be the pivot equation, since you can not turn the first elements in other rows to zero. Therefore, we need to switch the order of the equations to get a new pivot equation).

```
P, L, U = sp_la.lu(B)

print(P)
print(U)
print(L)
print(np.dot(L,U))
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
[[17.          2.          1.        ]
 [ 0.         11.88235294 -7.05882353]
 [ 0.          0.          3.01980198]]
[[ 1.          0.          0.        ]
 [ 0.05882353  1.          0.        ]
 [ 0.17647059 -0.11386139  1.        ]]
[[17.  2.  1.]
 [ 1. 12. -7.]
 [ 3. -1.  4.]]
```