Lab Problem: 04.

Course code: CCE-312.
Course Title: Numerical Methods sessional.

Name of the Lab Report: Solve Real world problem and Simul equation using Gauss-Jordan method.

Remarks & Signature:

Submitted To

Professor Dr. Md. Samsuzzaman.
Chairman,
Department of Computer and Communication Engineering.
Faculty of Computer Science & Engineering.

Submitted By

HASAN AHAMMAD
ID No: 1902073
Reg No: 08779
Level- 3, Semester- 1
Session: 2019-2020
Faculty of Computer Science & Engineering.
Patuakhali Science & Technology University.
Dumki, Patuakhali-8602.

1. **An investor has \$10,000 to invest in two types of financial instruments: stocks and bonds. The expected return from stocks is 8%, and from bonds is 5%. The investor wants the total return to be \$700. Find the amount invested in each type Using *Gauss-Jordan* method after that implement it using *Python.***

✪ Solve using Gauss-Jordan method.

from the Given problem we find two equations:-

0.08x+0.05y=700---(i)

x+y=10000---(ii)

here, x=stocks and y=bonds.

Converting given equations into matrix form

$$\left[\begin{array}{cc|c} 0.08 & 0.05 & 700 \\ 1 & 1 & 10000 \end{array}\right]$$

R_1<-R_1/0.08

$$\left[\begin{array}{cc|c} 1 & 00.625 & 8750 \\ 1 & 1 & 10000 \end{array}\right]$$

R_2<-R_2-R_1

$$\left[\begin{array}{cc|c} 1 & 00.625 & 8750 \\ 0 & 0.375 & 1250 \end{array}\right]$$

R_2<-R_2/.375

$$\left[\begin{array}{cc|c} 1 & 0.625 & 8750 \\ 0 & 1 & 3333.3333 \end{array}\right]$$

R_1<-R_1-.625xR_2

$$\left[\begin{array}{cc|c} 1 & 0 & 6666.667 \\ 0 & 1 & 3333.3333 \end{array}\right]$$

So, x=6666.667 and y=3333.3333.

Hence amount of stocks invested 6666 and amount of bonds invested 3333.

✪ **Implement using Python:**

```
import numpy as np

# Coefficients matrix
coefficients = np.array([[0.08, 0.05], [1, 1]])

# Constants vector
constants = np.array([700, 10000])

# Augmented matrix
augmented_matrix = np.column_stack((coefficients, constants))

# Applying Gauss-Jordan elimination
rows, cols = augmented_matrix.shape

for i in range(rows):
    # Normalize the pivot row
```

```
    augmented_matrix[i] = augmented_matrix[i] / augmented_matrix[i, i]

    # Eliminate other rows
    for j in range(rows):
        if i != j:
            augmented_matrix[j] = augmented_matrix[j] -
augmented_matrix[j, i] * augmented_matrix[i]

# Extract the solution
solution = augmented_matrix[:, -1]

# Print the solution
print("Amount invested in stocks:", solution[0])
print("Amount invested in bonds:", solution[1])
```

2. Solve the following system by the Gauss-Jordan method and Implement it using Python.

$3x_1-0.1x_2-0.2x_3=7.85$

$0.1x_1+7x_2-0.3x_3=-19.3$

$0.3x_1-0.2x_2+10x_3=71.4$

- **Step-01:** First we have to express the coefficients and the right-hand side as an augmented matrix:

$$\left[\begin{array}{ccc|c} 3 & -0.1 & -0.2 & 7.85 \\ 0.1 & 7 & -0.3 & -19.3 \\ 0.3 & -0.2 & 10 & 71.4 \end{array}\right]$$

- **Step-02:** Divide Row 1 by 3 to make the leading coefficient 1 in the first row:

$$\left[\begin{array}{ccc|c} 1 & -0.0333333 & -0.066667 & 2.61667 \\ 0.1 & 7 & -0.3 & -19.3 \\ 0.3 & -0.2 & 10 & 71.4 \end{array}\right] r_1'=r_1/3$$

- **Step-03:** Subtract 0.1 times Row 1 from Row 2 and 0.3 times Row 1 from Row 3 to make the entries below the leading 1 in Row 1 equal to 0:

$$\left[\begin{array}{ccc|c} 1 & -0.0333333 & -0.066667 & 2.61667 \\ 0 & 7.00333 & -0.29333 & -19.878 \\ 0 & -0.19000 & 10.0200 & 70.6150 \end{array}\right] r_2'=r_2-r_1x0.1 \text{ and } r_3'=r_3-r_1x0.3$$

- **Step-04:** Divide Row 2 by 7.0033 to make the leading coefficient 1 in the second row:

$$\left[\begin{array}{ccc|c} 1 & -0.0333333 & -0.066667 & 2.61667 \\ 0 & 1 & -0.0418848 & -2.79320 \\ 0 & -0.19000 & 10.0200 & 70.6150 \end{array}\right] r_2'=r_2/7.00333$$

- **Step-05:** Reduction of x_2 terms from first and third equation we can use,

$$\left[\begin{array}{ccc|c} 1 & 0 & -0.0680629 & 2.52356 \\ 0 & 1 & -0.0418848 & -2.79320 \\ 0 & 0 & 10.01200 & 70.0843 \end{array}\right] r_1'=r_1+r_2x0.0333 \text{ and } r_3'=r_3+r_2x0.1900$$

- **Step-06:** Divide Row 3 by 10.01200 to make the leading coefficient 1 in the third row:

$$\left[\begin{array}{ccc|c} 1 & 0 & -0.0680629 & 2.52356 \\ 0 & 1 & -0.0418848 & -2.79320 \\ 0 & 0 & 1 & 7 \end{array}\right] r_3'=r_3/10.01200$$

- **Step-07:** Finally, reducing x_3 terms from equation 1 and 2 we need,

$$\begin{bmatrix} 1 & 0 & 0 & 3.0 \\ 0 & 1 & 0 & -2.50 \\ 0 & 0 & 1 & 7.0 \end{bmatrix}$$ $r_1' = r_1 + r_3 x 0.0680629$ and $r_2' = r_2 + r_3 x 0.0418848$

Now, we find the value of x_1, x_2 and x_3,

$x_1 = 3.0$

$x_2 = -2.50$ and

$x_3 = 7.0$

- **Implement using python:**

```
import numpy as np

# Coefficients matrix
coefficients = np.array([[3, -0.1, -0.2], [0.1, 7, -0.3], [0.3, -0.2,
10]])

# Constants vector
constants = np.array([7.85, -19.3, 71.4])

# Augmented matrix
augmented_matrix = np.column_stack((coefficients, constants))

# Applying Gauss-Jordan elimination
rows, cols = augmented_matrix.shape

for i in range(rows):
    # Normalize the pivot row
    augmented_matrix[i] = augmented_matrix[i] / augmented_matrix[i,
i]

    # Eliminate other rows
    for j in range(rows):
        if i != j:
            augmented_matrix[j] = augmented_matrix[j] -
augmented_matrix[j, i] * augmented_matrix[i]

# Extract the solution
solution = augmented_matrix[:, -1]

# Print the solution
print("Solution:")
for i, value in enumerate(solution):
    print(f"x{i+1} = {value}")
```

3. **Using Gauss-Jordan method solve the following.**

$$.3x_1+.52x_2+x_3=0.01$$
$$.5x_1+.3x_2+.5x_3=.67$$
$$.1x_1+.3x_2+.5x_3=-.44$$

After that implement it using python.

Given equations:

$.3x_1+.52x_2+x_3=0.01$--(i)
$.5x_1+.3x_2+.5x_3=.67$--(ii)
$.1x_1+.3x_2+.5x_3=-.44$---(iii)

Converting given equations into matrix form

$$\left[\begin{array}{ccc|c} 0.3 & 0.52 & 1 & 0.01 \\ 0.5 & 0.3 & 0.5 & 0.67 \\ 0.1 & 0.3 & 0.5 & -0.44 \end{array}\right]$$

$$R_1 \leftarrow R_1 \div 0.3$$

$$= \left[\begin{array}{ccc|c} 1 & 1.7333 & 3.3333 & 0.0333 \\ 0.5 & 0.3 & 0.5 & 0.67 \\ 0.1 & 0.3 & 0.5 & -0.44 \end{array}\right]$$

$$R_2 \leftarrow R_2 - 0.5 \times R_1$$

$$= \left[\begin{array}{ccc|c} 1 & 1.7333 & 3.3333 & 0.0333 \\ 0 & -0.5667 & -1.1667 & 0.6533 \\ 0.1 & 0.3 & 0.5 & -0.44 \end{array}\right]$$

$$R_2 \leftarrow R_2 \div -0.5667$$

$$= \left[\begin{array}{ccc|c} 1 & 1.7333 & 3.3333 & 0.0333 \\ 0 & 1 & 2.0588 & -1.1529 \\ 0 & 0.1267 & 0.1667 & -0.4433 \end{array}\right]$$

$R_1 \leftarrow R_1 - 1.7333 \times R_2$

$$= \left[\begin{array}{ccc|c} 1 & 0 & -0.2353 & 2.0318 \\ 0 & 1 & 2.0588 & -1.1529 \\ 0 & 0.1267 & 0.1667 & -0.4433 \end{array}\right]$$

$R_3 \leftarrow R_3 - 0.1267 \times R_2$

$$= \left[\begin{array}{ccc|c} 1 & 0 & -0.2353 & 2.0318 \\ 0 & 1 & 2.0588 & -1.1529 \\ 0 & 0 & -0.0941 & -0.2973 \end{array}\right]$$

$R_3 \leftarrow R_3 \div -0.0941$

$$= \left[\begin{array}{ccc|c} 1 & 0 & -0.2353 & 2.0318 \\ 0 & 1 & 2.0588 & -1.1529 \\ 0 & 0 & 1 & 3.1587 \end{array}\right]$$

$R_1 \leftarrow R_1 + 0.2353 \times R_3$

$$= \left[\begin{array}{ccc|c} 1 & 0 & 0 & 2.775 \\ 0 & 1 & 2.0588 & -1.1529 \\ 0 & 0 & 1 & 3.1587 \end{array}\right]$$

$R_2 \leftarrow R_2 - 2.0588 \times R_3$

$$= \left[\begin{array}{ccc|c} 1 & 0 & 0 & 2.775 \\ 0 & 1 & 0 & -7.6562 \\ 0 & 0 & 1 & 3.1587 \end{array}\right]$$

$i.e.$

$x = 2.775$

$y = -7.6562$

$z = 3.1587$

Solution By Gauss jordan elimination method
$x = 2.775, y = -7.6562$ and $z = 3.1587$

✪ **Implement using Python:**

```
✪ import numpy as np

# Coefficients matrix
coefficients = np.array([[0.3, 0.52, 1], [0.5, 0.3, 0.5], [0.1, 0.3, 0.5]])

# Constants vector
constants = np.array([0.01, 0.67, -0.44])

# Augmented matrix
augmented_matrix = np.column_stack((coefficients, constants))

# Applying Gauss-Jordan elimination
rows, cols = augmented_matrix.shape

for i in range(rows):
    # Find the pivot row
    pivot_row = i
    for j in range(i + 1, rows):
        if abs(augmented_matrix[j, i]) > abs(augmented_matrix[pivot_row, i]):
            pivot_row = j

    # Swap the current row with the pivot row
    augmented_matrix[[i, pivot_row]] = augmented_matrix[[pivot_row, i]]

    # Normalize the pivot row
    augmented_matrix[i] = augmented_matrix[i] / augmented_matrix[i, i]

    # Eliminate other rows
    for j in range(rows):
        if i != j:
            augmented_matrix[j] = augmented_matrix[j] - augmented_matrix[j, i] *
augmented_matrix[i]

# Extract the solution
solution = augmented_matrix[:, -1]

# Print the solution
print("Solution:")
for i, value in enumerate(solution):
    print(f"x{i+1} = {value}")
```