2023

# ASSIGNMENT 4

# NUMERICAL METHODS (CS-406)

## GAUSS JORDAN IMPLEMENTATION (INVERSE OF MATRIX) IN PYTHON

SUBMITTED BY: UMANG KANCHAN

SUBMITTED TO: DR. AYESHA CHOUDHARY

```python
import numpy as np

# Function to invert a matrix using Gauss-Jordan elimination
def invert_matrix(A):
    n = A.shape[0]
    I = np.identity(n)
    B = np.concatenate((A, I), axis=1)

    # Applying Gauss-Jordan elimination
    for i in range(n):
        # Search for maximum in this column
        max_row = i
        for j in range(i + 1, n):
            if abs(B[j, i]) > abs(B[max_row, i]):
                max_row = j

        # Swap the maximum row with current row (column by column)
        for k in range(i, 2 * n):
            tmp = B[max_row, k]
            B[max_row, k] = B[i, k]
            B[i, k] = tmp

        # Divide the pivot row by the pivot element
        pivot = B[i, i]
        for k in range(i, 2 * n):
            B[i, k] /= pivot

        # Subtract the pivot row from all other rows
        for j in range(n):
            if j != i:
                factor = B[j, i]
                for k in range(i, 2 * n):
                    B[j, k] -= factor * B[i, k]

    # Extracting inverse of A from the augmented matrix
    A_inv = B[:, n:]

    return A_inv

# Get input from user
n = int(input("Enter size of square matrix: "))
A = np.zeros((n, n))
print("Enter matrix coefficients:")
for i in range(n):
    for j in range(n):
        A[i, j] = float(input("a[" + str(i) + "][" + str(j) + "] = "))

# Invert matrix using Gauss-Jordan elimination
A_inv = invert_matrix(A)

# Print solution
```

```
print("Inverse of matrix A:")
print(A_inv)
```

SAMPLE OUTPUT

Enter matrix coefficients:

a[0][0] = 1

a[0][1] = 2

a[0][2] = 1

a[1][0] = 2

a[1][1] = 3

a[1][2] = 4

a[2][0] = 1

a[2][1] = 5

a[2][2] = 0

Inverse of matrix A:

[[ 4.  -1.  -1. ]

 [-0.8  0.2  0.4]

 [-1.4  0.6  0.2]]