GAUSS JORDAN IMPLEMENTATION (INVERSE Of matrix) IN PYTHON

SUBMITTED BY: UMANG KANCHAN

SUBMITTED TO: Dr. AYESHA CHOUDHARY

ASSIGNMENT 4

NUMERICAL METHODS (CS-406)

2023

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]]