Lab1_Sol

January 31, 2025

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

0.0.1 Question 1: Create the below two matrices

[4]:
```
def create_matrices():
    A = np.array([[1, -1, 3], [5, 7, 9], [-4, 2, 8]])
    B = np.array([[5, 7, 4], [-1, 2, 5], [0, 8, 4]])
    return A, B
```

0.0.2 Question 2: Find $AB - B^2A$

[7]:
```
def compute_expression(A, B):
    AB = np.matmul(A, B)
    B2A = np.matmul(np.matmul(B, B), A)
    result = AB - B2A
    return result
```

0.0.3 (i) Display the second row and third column of $AB - B^2A$

0.0.4 (ii) Find the max and min entry of $AB - B^2A$

0.0.5 (iii) Compute the sum of the diagonal entries of $AB - B^2A$

[10]:
```
def analyze_matrix(matrix):
    second_row_third_column = matrix[1, 2]  # Second row, third column (0-based␣
↪indexing)
    max_entry = np.max(matrix)
    min_entry = np.min(matrix)
    trace = np.trace(matrix)
    return second_row_third_column, max_entry, min_entry, trace
```

0.0.6 Question 3: Show that $A^3 - 16A^2 + 70A - 228I = O$

[13]:
```
def check_matrix_identity(A):
    I = np.eye(A.shape[0])  # Identity matrix of same order as A
    lhs = np.linalg.matrix_power(A, 3) - 16 * np.linalg.matrix_power(A, 2) + 70␣
↪* A - 228 * I
```

```
    return np.allclose(lhs, np.zeros_like(A))  # Check if lhs is approximately␣
↪zero
```

0.0.7 Question 4: Write a Python function for matrix multiplication

[16]:
```
def matrix_multiplication(A, B):
    if A.shape[1] != B.shape[0]:
        return "Matrices are not compatible for multiplication."

    # Using NumPy's built-in function
    product_builtin = np.dot(A, B)

    # Without using built-in function (explicit implementation)
    product_explicit = np.zeros((A.shape[0], B.shape[1]))
    for i in range(A.shape[0]):
        for j in range(B.shape[1]):
            for k in range(A.shape[1]):
                product_explicit[i][j] += A[i][k] * B[k][j]

    return product_builtin, product_explicit
```

0.0.8 Execution

[19]:
```
A, B = create_matrices()
result_matrix = compute_expression(A, B)
row_col_value, max_val, min_val, trace_val = analyze_matrix(result_matrix)
identity_check = check_matrix_identity(A)
product_builtin, product_explicit = matrix_multiplication(A, B)
```

0.0.9 Display Results

[22]:
```
print("Matrix A:")
print(A)
print("\nMatrix B:")
print(B)
print("\nResult of AB - B²A:")
print(result_matrix)
print(f"\nSecond row, third column: {row_col_value}")
print(f"Max entry: {max_val}, Min entry: {min_val}")
print(f"Sum of diagonal entries: {trace_val}")
print(f"\nA³ - 16A² + 70A - 228I = 0: {identity_check}")
print("\nMatrix multiplication (Built-in function):")
print(product_builtin)
print("\nMatrix multiplication (Explicit implementation):")
print(product_explicit)
```

```
Matrix A:
[[ 1 -1  3]
```

```
 [ 5  7  9]
 [-4  2  8]]

Matrix B:
[[ 5  7  4]
 [-1  2  5]
 [ 0  8  4]]

Result of AB - B²A:
[[ -133  -662 -1340]
 [  -56  -197  -429]
 [  -30  -416  -830]]

Second row, third column: -429
Max entry: -30, Min entry: -1340
Sum of diagonal entries: -1160

A³ - 16A² + 70A - 228I = 0: True

Matrix multiplication (Built-in function):
[[  6  29  11]
 [ 18 121  91]
 [-22  40  26]]

Matrix multiplication (Explicit implementation):
[[  6.  29.  11.]
 [ 18. 121.  91.]
 [-22.  40.  26.]]
```

[ ]: