Presentation - Matgeo

Aryansingh Sonaye
AI25BTECH11032
EE1030 - Matrix Theory

September 26, 2025

Problem Statement

Problem 5.7.5
If

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \tag{1.1}$$

find A^2.

Description of Variables used

Variable	Value
A	$\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$
$\mathbf{u}$	$\begin{pmatrix} 1 \\ -1 \end{pmatrix}$

Table: Input variables

Theoretical Solution

Method 1: Direct Matrix Multiplication

$$
\begin{aligned}
A^2 &= A \cdot A && (2.1)\\
&= \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} && (2.2)\\
&= \begin{pmatrix} 1 \cdot 1 + (-1)(-1) & 1 \cdot (-1) + (-1)(1) \\ (-1)(1) + 1(-1) & (-1)(-1) + 1 \cdot 1 \end{pmatrix} && (2.3)\\
&= \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix} && (2.4)
\end{aligned}
$$

Theoretical Solution

Method 2: Vector–Matrix Representation

$$
\begin{aligned}
A &= \mathbf{u}\mathbf{u}^T && (2.5)\\
A^2 &= (\mathbf{u}\mathbf{u}^T)(\mathbf{u}\mathbf{u}^T) && (2.6)\\
&= \mathbf{u}(\mathbf{u}^T\mathbf{u})\mathbf{u}^T && (2.7)\\
\mathbf{u}^T\mathbf{u} &= 1^2 + (-1)^2 = 2 && (2.8)\\
\implies A^2 &= 2\mathbf{u}\mathbf{u}^T && (2.9)\\
&= 2A && (2.10)\\
&= \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix} && (2.11)
\end{aligned}
$$

Final Answer

$$A^2 = \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix} \tag{2.12}$$

Code - C

```
// All matrices are 2x2, flattened in row-major order:
// [a11, a12,
//  a21, a22]

void mat2_mul(const double *A, const double *B, double *C) {
    // C = A * B
    double a11 = A[0], a12 = A[1], a21 = A[2], a22 = A[3];
    double b11 = B[0], b12 = B[1], b21 = B[2], b22 = B[3];

    C[0] = a11*b11 + a12*b21; // c11
    C[1] = a11*b12 + a12*b22; // c12
    C[2] = a21*b11 + a22*b21; // c21
    C[3] = a21*b12 + a22*b22; // c22
}
```

Code - C

```
void mat2_square(const double *A, double *C) {
    // C = A^2
    mat2_mul(A, A, C);
}
```

Code - Python(with shared C code)

The code to obtain the required plot is

```
import ctypes
import numpy as np

# Load the shared library
lib = ctypes.CDLL("./libmat2.so")

# Define function prototypes
Nd = np.ctypeslib.ndpointer(dtype=np.double, ndim=1, flags="
    C_CONTIGUOUS")
lib.mat2_square.argtypes = [Nd, Nd]
lib.mat2_square.restype = None

# Define matrix A (flattened row-major)
A = np.array([1.0, -1.0,
              -1.0, 1.0], dtype=np.double)
```

Code - Python(with shared C code)

```
A2 = np.empty(4, dtype=np.double)

# Call C function: A2 = A^2
lib.mat2_square(A, A2)

# Reshape for printing
A_mat = A.reshape(2, 2)
A2_mat = A2.reshape(2, 2)

print("A =\n", A_mat)
print("\nA^2 (from C) =\n", A2_mat)
```

Code - Python only

```python
import numpy as np

# Define matrix A
A = np.array([[1, -1],
              [-1, 1]], dtype=float)

# Compute A^2
A2 = A @ A # or np.dot(A, A)

print("A =\n", A)
print("\nA^2 =\n", A2)
```