5.7.14

EE25BTECH11001 - Aarush Dilawri

October 3, 2025

Question:

If $\mathbf{A} = \begin{pmatrix} -3 & 6 \\ -2 & 4 \end{pmatrix}$, then show that $\mathbf{A}^3 = \mathbf{A}$.

Characteristic Polynomial

The characteristic equation of $\mathbf{A}$ is given by:

$$f(\lambda) = |\mathbf{A} - \lambda\mathbf{I}| = 0 \tag{1}$$

Therefore,

$$f(\lambda) = \begin{vmatrix} -3-\lambda & 6 \\ -2 & 4-\lambda \end{vmatrix} = 0 \tag{2}$$

$$f(\lambda) = \lambda^2 - \lambda = 0 \tag{3}$$

Applying Cayley-Hamilton

By Cayley-Hamilton theorem,

$$f\left(\lambda\right) = f\left(\mathbf{A}\right) = 0 \tag{4}$$

Therefore,

$$\mathbf{A}^2 - \mathbf{A} = 0 \implies \mathbf{A}^2 = \mathbf{A} \tag{5}$$

Final Step

Pre-multiplying both sides by $\mathbf{A}$,

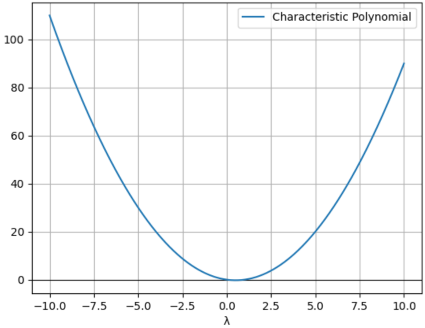
$$\mathbf{A}^3 = \mathbf{A}^2 \quad \text{but } \mathbf{A}^2 = \mathbf{A} \tag{6}$$

$$\implies \mathbf{A}^3 = \mathbf{A} \tag{7}$$

Hence proved.

Graphical Representation

See Figure,



C Code (code.c)

```
#include <stdio.h>

// Function to compute characteristic polynomial coefficients of a 2x2
    matrix
// Input: a11, a12, a21, a22
// Output: coeffs[0] = 1 (^2), coeffs[1] = -trace(A), coeffs[2] = det(A)
void char_poly(double a11, double a12, double a21, double a22, double
    * coeffs) {
    double trace = a11 + a22;
    double det = a11 * a22 - a12 * a21;

    coeffs[0] = 1.0; // ^2 coefficient
    coeffs[1] = -trace; //  coefficient
    coeffs[2] = det; // constant term
}
```

Python Code (code.py)

```
import numpy as np
import matplotlib.pyplot as plt

# Given matrix A
a11, a12, a21, a22 = -3, 6, -2, 4

# Compute trace and determinant
trace = a11 + a22
det = a11*a22 - a12*a21

# Polynomial coefficients
coeffs = [1, -trace, det]
print("Characteristic Polynomial Coefficients:", coeffs)
```

Python Code (code.py)

```
# Define polynomial
lam = np.linspace(−10, 10, 400)
poly_vals = coeffs[0]∗lam∗∗2 + coeffs[1]∗lam + coeffs[2]

# Plot
plt.axhline(0, color='black', linewidth=0.8)
plt.plot(lam, poly_vals, label="Characteristic Polynomial")
plt.xlabel("")
plt.ylabel("p()")
plt.legend()
plt.grid(True)
plt.show()
```

Python Code (nativecode.py)

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt

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

# Define argument and return types
code.char_poly.argtypes = [ctypes.c_double, ctypes.c_double, ctypes.
    c_double, ctypes.c_double,
                                  np.ctypeslib.ndpointer(dtype=np.float64,
                                       ndim=1, flags="C")]

coeffs = np.zeros(3, dtype=np.float64)

a11, a12, a21, a22 = -3, 6, -2, 4
```

Python Code (nativecode.py)

```
# Call C function
code.char_poly(a11, a12, a21, a22, coeffs)
print("Characteristic Polynomial Coefficients:", coeffs)

# Define polynomial
lam = np.linspace(-10, 10, 400)
poly_vals = coeffs[0]*lam**2 + coeffs[1]*lam + coeffs[2]

# Plot
plt.axhline(0, color='black', linewidth=0.8)
plt.plot(lam, poly_vals, label="Characteristic Polynomial")
plt.xlabel("")
plt.ylabel("p()")
plt.legend()
plt.grid(True)
plt.show()
```