Presentation - Matgeo

Aryansingh Sonaye
AI25BTECH11032
EE1030 - Matrix Theory

September 30, 2025

Problem Statement

Problem 12.214

The eigenvector pair of the matrix

$$A = \begin{pmatrix} 3 & 4 \\ 4 & -3 \end{pmatrix} \tag{1.1}$$

is (PI 2008)

Options:

$$\text{(a)} \quad \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -2 \end{pmatrix} \tag{1.2}$$

$$\text{(b)} \quad \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -1 \end{pmatrix} \tag{1.3}$$

$$\text{(c)} \quad \begin{pmatrix} 1 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -1 \end{pmatrix} \tag{1.4}$$

$$\text{(d)} \quad \begin{pmatrix} 1 \\ -2 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix} \tag{1.5}$$

Description of Variables used

Symbol	Description
A	Given matrix $\begin{pmatrix} 3 & 4 \\ 4 & -3 \end{pmatrix}$
λ	Eigenvalue of A
$\mathbf{v}$	Corresponding eigenvector

Theoretical Solution

$$A = \begin{pmatrix} 3 & 4 \\ 4 & -3 \end{pmatrix} \tag{2.1}$$

$$\det(A - \lambda I) = \det \begin{pmatrix} 3 - \lambda & 4 \\ 4 & -3 - \lambda \end{pmatrix} \tag{2.2}$$

$$= (3 - \lambda)(-3 - \lambda) - 16 \tag{2.3}$$

$$= \lambda^2 - 25 \tag{2.4}$$

$$\Rightarrow \lambda = \pm 5 \tag{2.5}$$

Theoretical Solution

For $\lambda = 5$:

$$(A - 5I)\mathbf{v} = 0 \tag{2.6}$$

$$\begin{pmatrix} -2 & 4 \\ 4 & -8 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{0} \tag{2.7}$$

$$-2x + 4y = 0 \Rightarrow x = 2y \tag{2.8}$$

$$\mathbf{v_1} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \tag{2.9}$$

For $\lambda = -5$:

$$(A + 5I)\mathbf{v} = 0 \tag{2.10}$$

$$\begin{pmatrix} 8 & 4 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{0} \tag{2.11}$$

$$8x + 4y = 0 \Rightarrow y = -2x \tag{2.12}$$

Theoretical Solution

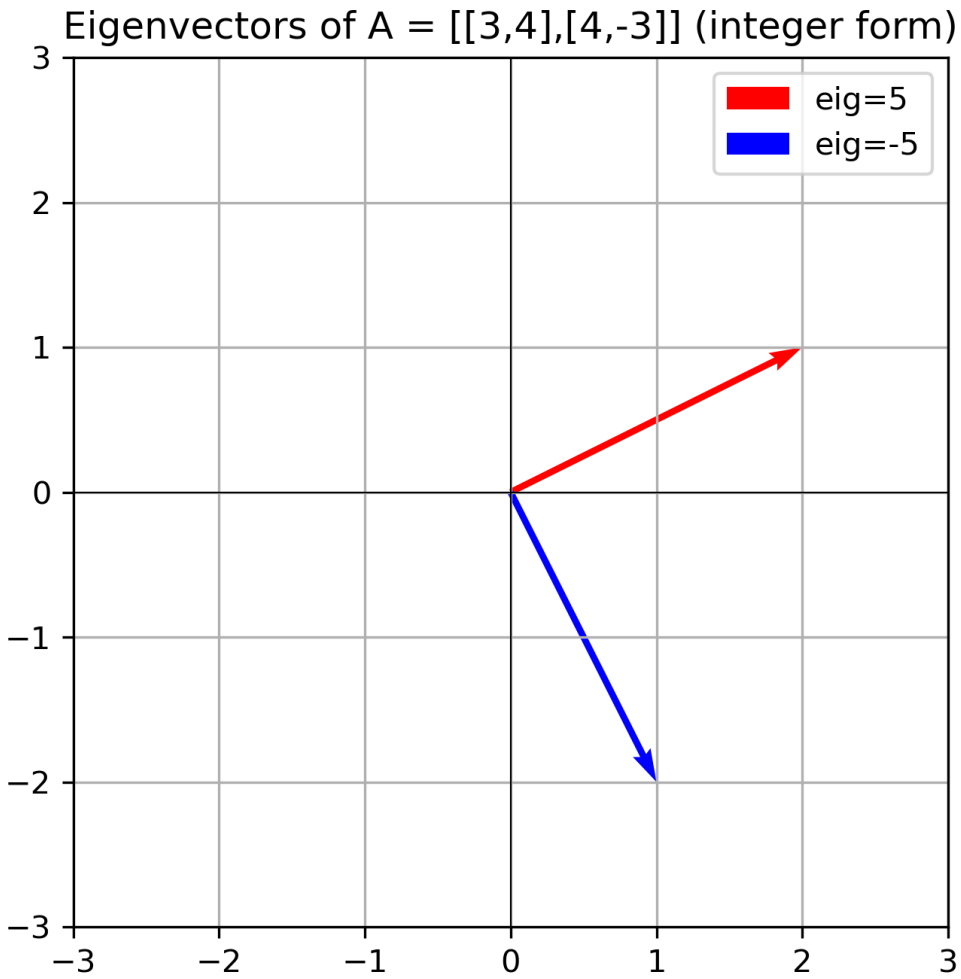
$$\mathbf{v_2} = \begin{pmatrix} 1 \\ -2 \end{pmatrix} \tag{2.14}$$

Hence, the correct eigenvector pair is

$$\mathbf{v_1} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad \mathbf{v_2} = \begin{pmatrix} 1 \\ -2 \end{pmatrix}. \tag{2.15}$$

Answer: (a)

Plot


Eigenvectors of A = [[3,4],[4,-3]] (integer form)
eig=5
eig=-5
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Figure

Code - C

```c
#include <stdio.h>
#include <math.h>

// Compute eigenvalues of a 2x2 real matrix
// A = [a b; c d]
void eigenvalues(double a, double b, double c, double d, double *eig1,
    double *eig2) {
    double trace = a + d;
    double det = a * d - b * c;
    double disc = trace * trace - 4 * det;

    if (disc < 0) { // complex case not handled
        *eig1 = *eig2 = NAN;
        return;}
    *eig1 = (trace + sqrt(disc)) / 2.0;
    *eig2 = (trace - sqrt(disc)) / 2.0;
}
```

Code - C

```
// Compute eigenvector for given eigenvalue
void eigenvector(double a, double b, double c, double d, double eig,
    double *v) {
    double m11 = a - eig;
    double m12 = b;
    double m21 = c;
    double m22 = d - eig;
    // Try first row equation: m11*x + m12*y = 0
    if (fabs(m11) > 1e-6 || fabs(m12) > 1e-6) {
        if (fabs(m12) > 1e-6) {
            v[0] = 1.0;
            v[1] = -m11 / m12;
        } else {
            v[0] = -m12 / m11;
            v[1] = 1.0;
        }
    }
```

Code - C

```
    // Otherwise try second row
    else if (fabs(m21) > 1e-6 || fabs(m22) > 1e-6) {
        if (fabs(m22) > 1e-6) {
            v[0] = 1.0;
            v[1] = -m21 / m22;
        } else {
            v[0] = -m22 / m21;
            v[1] = 1.0;
        }
    } else {
        v[0] = v[1] = 0.0; // degenerate case
    }
}
```

Code - Python(with shared C code)

The code to obtain the required plot is

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

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

# Define arg/return types
lib.eigenvalues.argtypes = [ctypes.c_double, ctypes.c_double,
                            ctypes.c_double, ctypes.c_double,
                            ctypes.POINTER(ctypes.c_double),
                            ctypes.POINTER(ctypes.c_double)]
lib.eigenvalues.restype = None
```

Code - Python(with shared C code)

```
lib.eigenvector.argtypes = [ctypes.c_double, ctypes.c_double,
                            ctypes.c_double, ctypes.c_double,
                            ctypes.c_double,
                            ctypes.POINTER(ctypes.c_double)]
lib.eigenvector.restype = None

# Example matrix A = [[3,4],[4,-3]]
a, b, c, d = 3.0, 4.0, 4.0, -3.0

eig1 = ctypes.c_double()
eig2 = ctypes.c_double()

# Get eigenvalues
lib.eigenvalues(a, b, c, d, ctypes.byref(eig1), ctypes.byref(eig2))
print("Eigenvalues:", eig1.value, eig2.value)
```

Code - Python(with shared C code)

```
vec1 = np.array([v1[0], v1[1]])
vec2 = np.array([v2[0], v2[1]])

print("Eigenvector for eig1:", vec1)
print("Eigenvector for eig2:", vec2)

# ----- Plot -----
plt.figure(figsize=(5,5))
plt.axhline(0, color="black", linewidth=0.5)
plt.axvline(0, color="black", linewidth=0.5)

plt.quiver(0, 0, vec1[0], vec1[1], angles='xy', scale_units='xy', scale=1,
           color="red", label=f"eig1={eig1.value:.2f}")
plt.quiver(0, 0, vec2[0], vec2[1], angles='xy', scale_units='xy', scale=1,
           color="blue", label=f"eig2={eig2.value:.2f}")
```

Code - Python(with shared C code)

```
plt.xlim(−3, 3)
plt.ylim(−3, 3)
plt.grid(True)
plt.legend()
plt.title("Eigenvectors of A = [[3,4],[4,−3]]")

# Save figure
plt.savefig("eigenvectors.png", dpi=300)

# Show figure
plt.show()
```

Code - Python only

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

# Define the matrix
A = np.array([[3, 4],
              [4, -3]], dtype=float)
# Compute eigenvalues and eigenvectors
eigvals, eigvecs = np.linalg.eig(A)
print("Matrix A:\n", A)
print("\nEigenvalues:", eigvals)
print("\nEigenvectors (normalized, from NumPy):\n", eigvecs)

# ----- Force integer eigenvectors -----
# For eig = +5, eigenvector ~ (2,1)
v1 = np.array([2, 1], dtype=float)
# For eig = -5, eigenvector ~ (1,-2)
v2 = np.array([1, -2], dtype=float)
```

Code - Python only

```
print("\nAdjusted Eigenvectors:")
print("Eigenvalue 5 :", v1)
print("Eigenvalue −5 :", v2)

# ———— Plot ————
plt.figure(figsize=(5,5))
plt.axhline(0, color="black", linewidth=0.5)
plt.axvline(0, color="black", linewidth=0.5)

plt.quiver(0, 0, v1[0], v1[1], angles='xy', scale_units='xy', scale=1,
           color="red", label="eig=5")
plt.quiver(0, 0, v2[0], v2[1], angles='xy', scale_units='xy', scale=1,
           color="blue", label="eig=−5")
```

Code - Python only

```
plt.xlim(−3, 3)
plt.ylim(−3, 3)
plt.grid(True)
plt.legend()
plt.title("Eigenvectors of A = [[3,4],[4,−3]] (integer form)")

# Save and show
plt.savefig("eigenvectors_problem_adjusted.png", dpi=300)
plt.show()
```