12.41

INDHIRESH S - EE25BTECH11027

7 October, 2025

Question

The matrix $A = \begin{pmatrix} 4 & 3 \\ 9 & -2 \end{pmatrix}$ has eigenvalues -5 and 7. The eigenvector(s) is/are

1. $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$
2. $\begin{pmatrix} 3 \\ 4 \end{pmatrix}$
3. $\begin{pmatrix} 2 \\ -6 \end{pmatrix}$
4. $\begin{pmatrix} 2 \\ 8 \end{pmatrix}$

Given

$$\mathbf{A} = \begin{pmatrix} 4 & 3 \\ 9 & -2 \end{pmatrix} \ , \lambda_1 = -5 \ \ \textit{and} \ \ \lambda_2 = 7 \tag{1}$$

Now

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \tag{2}$$

$$\left(\mathbf{A} - \lambda\mathbf{I}\right)\mathbf{x} = 0 \tag{3}$$

Here **x** is eigen vector.

$$\left(\begin{pmatrix} 4 & 3 \\ 9 & -2 \end{pmatrix} - \lambda\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right)\mathbf{x} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{4}$$

Theoretical solution

For $\lambda = -5$

$$\begin{pmatrix} 9 & 3 \\ 9 & 3 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (5)$$

Let

$$\mathbf{x} = \begin{pmatrix} a \\ b \end{pmatrix} \quad (6)$$

By substituting Eq.6 in Eq.5 we get

$$9a = -3b \quad (7)$$

$$3a = -b \quad (8)$$

Theoretical solution

$$\mathbf{x} = \begin{pmatrix} a \\ -3a \end{pmatrix} \tag{9}$$

$$\mathbf{x} = a\begin{pmatrix} 1 \\ -3 \end{pmatrix} \tag{10}$$

Where a is a scalar. So the eigen vector will be the scalar multiple of $\mathbf{x}$.
For $a = 2$

$$\mathbf{x} = \begin{pmatrix} 2 \\ -6 \end{pmatrix} \tag{11}$$

Option (3) is correct

Theoretical solution

For $\lambda = 7$

$$\begin{pmatrix} -3 & 3 \\ 9 & -9 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{12}$$

By substituting Eq.6 in Eq.11 we get

$$a = b \tag{13}$$

$$\mathbf{x} = \begin{pmatrix} a \\ a \end{pmatrix} \tag{14}$$

$$\mathbf{x} = a \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{15}$$

Where a is a scalar. So the eigen vector will be the scalar multiple of **x**.

Theoretical solution

For $a = 1$

$$\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{16}$$

Option (1) is also correct

```c
#include <math.h>
int find_common_tangents(double x1, double y1, double r1, double
    x2, double y2, double r2) {
    // Calculate the distance between the centers
    double d = sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));

    // Calculate the sum and difference of the radii
    double r_sum = r1 + r2;
    double r_diff = fabs(r1 - r2);

    // Determine the relationship between the circles
    if (d > r_sum) {
        return 4; // Circles are separate and do not intersect
    } else if (d == r_sum) {
        return 3; // Circles touch externally
    } else if (d > r_diff && d < r_sum) {
        return 2; // Circles intersect at two points
    }
```

C Code

```
    else if (d == r_diff) {
        return 1; // Circles touch internally
    }
    else if (d < r_diff) {
        return 0; // One circle is completely inside the other
    } else if (d == 0 && r1 == r2) {
        return -1; // Concentric and identical
    }

    return 0; // Default case, including d=0 and r1!=r2
}
```

Python Code

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

# --- 1. Load the C shared library ---
lib_name = 'circle.so'
if platform.system() == 'Windows':
    lib_name = 'circle.dll'

try:
    c_lib = ctypes.CDLL(f'./{lib_name}')
except OSError as e:
    print(fError loading shared library: {e})
    print(fPlease compile circle.c into {lib_name} first.)
    exit()
```

Python Code

```
 # --- 2. Define the C function signature for Python ---
c_lib.find_common_tangents.argtypes = [
    ctypes.c_double, ctypes.c_double, ctypes.c_double,
    ctypes.c_double, ctypes.c_double, ctypes.c_double
]
c_lib.find_common_tangents.restype = ctypes.c_int

def solve_with_c(c1, r1, c2, r2):
    A Python wrapper that calls the C function.
    return c_lib.find_common_tangents(c1[0], c1[1], r1, c2[0], c2
        [1], r2)

def plot_circles(c1, r1, c2, r2, tangency_point):
    Plots the two circles, their centers, and the point of
        tangency.
    fig, ax = plt.subplots(figsize=(10, 8))
```

Python Code

```
 # Create circle patches
circle1 = plt.Circle(c1, r1, color='blue', fill=False,
    linewidth=2, label=f'Circle 1: $r_1={r1}$')
circle2 = plt.Circle(c2, r2, color='red', fill=False,
    linewidth=2, label=f'Circle 2: $r_2={r2}$')

ax.add_patch(circle1)
ax.add_patch(circle2)

# Plot centers and label them with coordinates
ax.plot(c1[0], c1[1], 'bo', markersize=8, label='Center $C_1$
    ')
ax.text(c1[0] + 0.3, c1[1] + 0.3, f'$C_1$ ({c1[0]:.1f}, {c1
    [1]:.1f})', fontsize=12, color='blue')
```

```python
    ax.plot(c2[0], c2[1], 'ro', markersize=8, label='Center $C_2$')
    ax.text(c2[0] + 0.3, c2[1] + 0.3, f'$C_2$ ({c2[0]:.1f}, {c2[1]:.1f})', fontsize=12, color='red')

    # Plot tangency point and label it with coordinates
    ax.plot(tangency_point[0], tangency_point[1], 'go', markersize=8, label='Tangency Point T')
    ax.text(tangency_point[0] + 0.3, tangency_point[1] - 0.5, f'T ({tangency_point[0]}, {tangency_point[1]})', fontsize=12, color='green')

    # Set plot properties
    ax.set_aspect('equal', adjustable='box')
    plt.title('Relationship Between Two Circles (Internal Tangency)')
```

```
    plt.xlabel('x-axis')
    plt.ylabel('y-axis')
    plt.grid(True)
    plt.legend(loc='upper right')
    plt.savefig(/media/indhiresh-s/New Volume/Matrix/ee1030-2025/
        ee25btech11027/MATGEO/10.7.76/figs/figure1.png)
    plt.show()

# --- Main execution ---
if __name__ == __main__:
    # Parameters for the circles from the problem
    C1 = (0.0, 0.0)
    r1 = 2.0
    C2 = (3.0, 4.0)
    r2 = 7.0
```

Python Code

```
# Calculate the number of tangents using the C function
num_tangents = solve_with_c(C1, r1, C2, r2)
print(fNumber of common tangents (via C function): {
    num_tangents})

# Calculate the point of tangency for plotting
v = np.array(C1) - np.array(C2)
u = v / np.linalg.norm(v)
T = tuple(np.array(C2) + r2 * u)
T_rounded = (round(T[0], 2), round(T[1], 2))

# Plot the result
plot_circles(C1, r1, C2, r2, T_rounded)
```

Plot

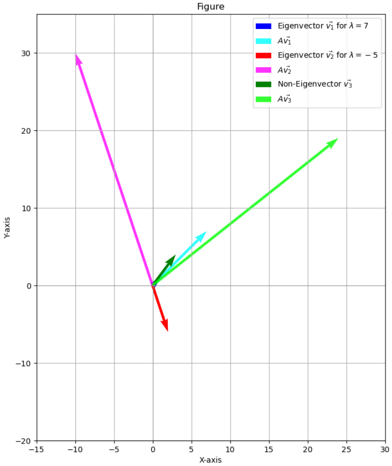

Figure
Eigenvector $\vec{v_1}$ for $\lambda = 7$
$A\vec{v_1}$
Eigenvector $\vec{v_2}$ for $\lambda = -5$
$A\vec{v_2}$
Non-Eigenvector $\vec{v_3}$
$A\vec{v_3}$
30
20
10
0
−10
−20
Y-axis
−15
−10
−5
0
5
10
15
20
25
30
X-axis