5.7.8

AI25BTECH11035-SUJAL RAJANI

September 5,2025

Question

Given that $\mathbf{A} = \begin{pmatrix} \alpha & \beta \\ \gamma & -\alpha \end{pmatrix}$ and $\mathbf{A}^2 = 3\mathbf{I}$, then

(a) $1 + \alpha^2 + \beta\gamma = 0$
(b) $1 - \alpha^2 - \beta\gamma = 0$
(c) $3 - \alpha^2 - \beta\gamma = 0$
(d) $3 + \alpha^2 + \beta\gamma = 0$

Theoretical Solution

as mentioned in the question :

$$\mathbf{A} = \begin{pmatrix} \alpha & \beta \\ \gamma & -\alpha \end{pmatrix}$$

$$\mathbf{A}^2 = 3\mathbf{I}. \tag{1}$$

The characteristic equation of **A** is:

$$\left| \begin{pmatrix} \alpha - \lambda & \beta \\ \gamma & -\alpha - \lambda \end{pmatrix} \right| = 0, \tag{2}$$

which expands to

$$\lambda^2 - \alpha^2 - \beta\gamma = 0. \tag{3}$$

Theoretical Solution

By Carlyle-Hamilton theorem

$$\mathbf{A}^2 = (\alpha^2 + \beta\gamma)\mathbf{I}$$

by equation 1 :

$$3\mathbf{I} = (\alpha^2 + \beta\gamma)\mathbf{I} \tag{4}$$

$$3 = \alpha^2 + \beta\gamma$$

. option c is the correct option .

C Code -Finding Solution for the system of Equations

```c
#include <stdio.h>

void rref_solver(double aug[2][3], double solution[2]) {
    // Normalize first row (pivot = aug[0][0])
    double pivot = aug[0][0];
    for (int j = 0; j < 3; j++) {
        aug[0][j] /= pivot;
    }
    // Eliminate below pivot
    double factor = aug[1][0];
    for (int j = 0; j < 3; j++) {
        aug[1][j] -= factor * aug[0][j];
    }
```

C Code -Finding Solution for the system of Equations

```c
    // Normalize second row (pivot = aug[1][1])
    pivot = aug[1][1];
    for (int j = 0; j < 3; j++) {
        aug[1][j] /= pivot;
    }
    // Eliminate above pivot
    factor = aug[0][1];
    for (int j = 0; j < 3; j++) {
        aug[0][j] -= factor * aug[1][j];
    }
    // Extract solution
    solution[0] = aug[0][2]; // x
    solution[1] = aug[1][2]; // y
}
```

Python+C code

```python
import ctypes
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mp
mp.use("TkAgg")
# Load the shared C library
lib = ctypes.CDLL("./liblineq_solver.so")
# Define argument and return types
lib.rref_solver.argtypes = [ctypes.c_double * 6, ctypes.c_double
    * 2]
# Create augmented matrix for system:
aug = (ctypes.c_double * 6)(1, 2, 6, 2, -5, 12) # Flattened 2x3
solution = (ctypes.c_double * 2)()
lib.rref_solver(aug, solution)
# Convert result to numpy vector (ensure flat)
x_sol = np.array([solution[0], solution[1]], dtype=float).flatten
    ()
print("Solution vector from C:", x_sol)
```

Python+C code

```
# plot
x_vals = np.linspace(-2, 10, 400)
y1 = (6 - x_vals) / 2
y2 = (12 - 2*x_vals) / -5

plt.plot(x_vals, y1, label=r"$x+3y=6$")
plt.plot(x_vals, y2, label=r"$2x-3y=12$")

plt.scatter(x_sol[0], x_sol[1], color="red", zorder=5)
plt.text(float(x_sol[0])+0.2, float(x_sol[1]), f"({x_sol[0]:.1f},
    {x_sol[1]:.1f})", color="red")
```

Python+C code

```
plt.xlabel("x")
plt.ylabel("y")
plt.title("Garphical solution of the Linear system")
plt.axhline(0, color="black", linewidth=0.8)
plt.axvline(0, color="black", linewidth=0.8)
plt.legend()
plt.grid(True)
plt.show()
```

Python code

```python
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mp
mp.use("TkAgg")

A=np.array([[1,2],[2,-5]],dtype=float)
b=np.array([6,12], dtype=float)

x=np.linalg.solve(A,b)
print("Solution vector for the system of equations:",x)
```

Python code

```python
# Making a plot
x_vals = np.linspace(-2, 10, 400)

# Rearranged equations to express y in terms of x
y1 = (6 - x_vals) / 2 # from x + 3y = 6
y2 = (12 - 2*x_vals) / -5 # from 2x - 3y = 12

# Plot lines
plt.plot(x_vals, y1, label=r"$x + 2y = 6$")
plt.plot(x_vals, y2, label=r"$2x - 5y = 12$")

# Mark solution
plt.scatter(x[0], x[1], color="red", zorder=5)
plt.text(x[0]+0.2, x[1], f"({x[0]:.1f}, {x[1]:.1f})", color="red"
    )
```

Python code

```
# Formatting
plt.xlabel("x")
plt.ylabel("y")
plt.title("Graphical Solution of the Linear System")
plt.axhline(0, color='black', linewidth=0.8)
plt.axvline(0, color='black', linewidth=0.8)
plt.legend()
plt.grid(True)
plt.show()
```