5.13.38

BEERAM MADHURI - EE25BTECH11012

October 2025

Question

Let $\mathbf{X}$ and $\mathbf{Y}$ be two arbitrary, 3×3, non-zero, skew-symmetric matrices and $\mathbf{Z}$ be an arbitrary 3×3, non-zero, symmetric matrix. Then which of the following matrices is (are) skew symmetric?

1. $\mathbf{Y}^3\mathbf{Z}^4 - \mathbf{Z}^4\mathbf{Y}^3$
2. $\mathbf{X}^{44} + \mathbf{Y}^{44}$
3. $\mathbf{X}^4\mathbf{Z}^3 - \mathbf{Z}^3\mathbf{X}^4$
4. $\mathbf{X}^{23} + \mathbf{Y}^{23}$

finding Skew Symmetric Matrices:

Given,

$$
\begin{aligned}
\mathbf{X}^\top &= -\mathbf{X} \quad \text{(Skew-Symmetric)} \\
\mathbf{Y}^\top &= -\mathbf{Y} \quad \text{(Skew-Symmetric)} \\
\mathbf{Z}^\top &= \mathbf{Z} \quad \text{(Symmetric)}
\end{aligned} \tag{1}
$$

Checking all options:
a) $\mathbf{Y}^3\mathbf{Z}^4 - \mathbf{Z}^4\mathbf{Y}^3$
Let

$$A = \mathbf{Y}^3\mathbf{Z}^4 - \mathbf{Z}^4\mathbf{Y}^3 \tag{2}$$

$$A^\top = (\mathbf{Y}^3\mathbf{Z}^4 - \mathbf{Z}^4\mathbf{Y}^3)^\top \tag{3}$$

$$A^\top = (\mathbf{Y}^3\mathbf{Z}^4)^\top - (\mathbf{Z}^4\mathbf{Y}^3)^\top \tag{4}$$

$$A^\top = (\mathbf{Z}^\top)^4(\mathbf{Y}^\top)^3 - (\mathbf{Y}^\top)^3(\mathbf{Z}^\top)^4 \tag{5}$$

$$A^\top = (\mathbf{Z}^\top)^4(\mathbf{Y}^\top)^3 - (\mathbf{Y}^\top)^3(\mathbf{Z}^\top)^4 \tag{6}$$

Now, substitute the given properties ($\mathbf{Z}^\top = \mathbf{Z}$ and $\mathbf{Y}^\top = -\mathbf{Y}$):

$$A^\top = (\mathbf{Z})^4(-\mathbf{Y})^3 - (-\mathbf{Y})^3(\mathbf{Z})^4 \tag{7}$$

$$\text{Since } (-\mathbf{Y})^3 = (-1)^3\mathbf{Y}^3 = -\mathbf{Y}^3, \text{ we get:} \tag{8}$$

$$A^\top = \mathbf{Z}^4(-\mathbf{Y}^3) - (-\mathbf{Y}^3)\mathbf{Z}^4 \tag{9}$$

$$A^\top = -\mathbf{Z}^4\mathbf{Y}^3 + \mathbf{Y}^3\mathbf{Z}^4 \tag{10}$$

$$A^\top = \mathbf{Y}^3\mathbf{Z}^4 - \mathbf{Z}^4\mathbf{Y}^3 \tag{11}$$

$$= \mathbf{A} \tag{12}$$

Hence, A is Symmetric Matrix.

b) $\mathbf{X}^{44} + \mathbf{Y}^{44}$

Let

$$B = \mathbf{X}^{44} + \mathbf{Y}^{44} \tag{13}$$

$$B^{\top} = (\mathbf{X}^{44} + \mathbf{Y}^{44})^{\top} \tag{14}$$

$$B^{\top} = (\mathbf{X}^{44})^{\top} + (\mathbf{Y}^{44})^{\top} \tag{15}$$

$$B^{\top} = (\mathbf{X}^{\top})^{44} + (\mathbf{Y}^{\top})^{44} \tag{16}$$

Now, substitute the given properties ($\mathbf{X}^\top = -\mathbf{X}$ and $\mathbf{Y}^\top = -\mathbf{Y}$):

$$B^\top = (-\mathbf{X})^{44} + (-\mathbf{Y})^{44} \tag{17}$$

$$(-1)^{44} = 1 \tag{18}$$

$$\therefore (-\mathbf{X})^{44} = \mathbf{X}^{44} and (-\mathbf{Y})^{44} = \mathbf{Y}^{44} \tag{19}$$

$$B^\top = \mathbf{X}^{44} + \mathbf{Y}^{44} \tag{20}$$

$$B^\top = B \tag{21}$$

Hence, B is Symmetric Matrix.

c) $\mathbf{X}^4\mathbf{Z}^3 - \mathbf{Z}^3\mathbf{X}^4$

Let

$$C = (\mathbf{X}^4\mathbf{Z}^3 - \mathbf{Z}^3\mathbf{X}^4) \tag{22}$$

$$C^\top = (\mathbf{X}^4\mathbf{Z}^3 - \mathbf{Z}^3\mathbf{X}^4)^\top \tag{23}$$

$$C^\top = (\mathbf{X}^4\mathbf{Z}^3)^\top - (\mathbf{Z}^3\mathbf{X}^4)^\top \tag{24}$$

$$C^\top = (\mathbf{Z}^3)^\top(\mathbf{X}^4)^\top - (\mathbf{X}^4)^\top(\mathbf{Z}^3)^\top \tag{25}$$

$$C^\top = (\mathbf{Z}^\top)^3(\mathbf{X}^\top)^4 - (\mathbf{X}^\top)^4(\mathbf{Z}^\top)^3 \tag{26}$$

Now, substitute the given properties ($\mathbf{Z}^\top = \mathbf{Z}$ and $\mathbf{X}^\top = -\mathbf{X}$):

$$C^\top = (\mathbf{Z})^3(-\mathbf{X})^4 - (-\mathbf{X})^4(\mathbf{Z})^3 \tag{27}$$

$$(-\mathbf{X})^4 = \mathbf{X}^4 \tag{28}$$

$$\mathbf{C}^\top = \mathbf{Z}^3\mathbf{X}^4 - \mathbf{X}^4\mathbf{Z}^3 \tag{29}$$

$$C^\top = -(\mathbf{X}^4\mathbf{Z}^3 - \mathbf{Z}^3\mathbf{X}^4) \tag{30}$$

$$= -C \tag{31}$$

Hence, C is Skew Symmetric Matrix.

d) $\mathbf{X}^{23} + \mathbf{Y}^{23}$

Let

$$D = \mathbf{X}^{23} + \mathbf{Y}^{23} \tag{32}$$

$$D^\top = (\mathbf{X}^{23} + \mathbf{Y}^{23})^\top \tag{33}$$

$$D^\top = (\mathbf{X}^{23})^\top + (\mathbf{Y}^{23})^\top \tag{34}$$

$$D^\top = (\mathbf{X}^\top)^{23} + (\mathbf{Y}^\top)^{23} \tag{35}$$

Now, substitute the given properties ($\mathbf{X}^\top = -\mathbf{X}$ and $\mathbf{Y}^\top = -\mathbf{Y}$):

$$D^\top = (-\mathbf{X})^{23} + (-\mathbf{Y})^{23} \tag{36}$$

$$(-1)^{23} = -1 \tag{37}$$

$$(-\mathbf{X})^{23} = -\mathbf{X}^{23} \text{ and } (-\mathbf{Y})^{23} = -\mathbf{Y}^{23} \tag{38}$$

$$D^\top = -\mathbf{X}^{23} - \mathbf{Y}^{23} \tag{39}$$

$$D^\top = -(\mathbf{X}^{23} + \mathbf{Y}^{23}) \tag{40}$$

$$= -D \tag{41}$$

Hence, D is Skew Symmetric Matrix.

$\therefore$ Option 3. $\mathbf{X}^4\mathbf{Z}^3 - \mathbf{Z}^3\mathbf{X}^4$ and
Option 4. $\mathbf{X^{23}} + \mathbf{Y}^{23}$ are Skew Symmetric

Python Code

```python
import numpy as np

# --- 1. Define Helper Functions ---
def is_skew_symmetric(matrix):
    """
    Checks if a matrix M is skew-symmetric by verifying if M^T =
        -M.
    Uses np.allclose for accurate floating-point comparisons.
    """
    return np.allclose(matrix.T, -matrix)
```

python Code

```python
def is_symmetric(matrix):
    """Checks if a matrix M is symmetric by verifying if M^T = M.
    """
    return np.allclose(matrix.T, matrix)
# --- 2. Generate Arbitrary Non-Zero Matrices ---
# To create a skew-symmetric matrix, we can use the formula A - A
    ^T
# To create a symmetric matrix, we can use the formula A + A^T

# Create a random 3x3 matrix to generate X
A = np.random.rand(3, 3)
X = A - A.T # X is now skew-symmetric
```

python Code

```python
# Create another random 3x3 matrix to generate Y
B = np.random.rand(3, 3)
Y = B - B.T # Y is now skew-symmetric

# Create a random 3x3 matrix to generate Z
C = np.random.rand(3, 3)
Z = C + C.T # Z is now symmetric
```

python Code

```python
# --- 3. Verify Properties of Generated Matrices ---
print("--- Initial Matrix Properties ---")
print(f"Is X skew-symmetric? {is_skew_symmetric(X)}")
print(f"Is Y skew-symmetric? {is_skew_symmetric(Y)}")
print(f"Is Z symmetric? {is_symmetric(Z)}")
print("-" * 35)

# Note: @ is the operator for matrix multiplication in NumPy
M_a = np.linalg.matrix_power(Y, 3) @ np.linalg.matrix_power(Z, 4)
    - \
     np.linalg.matrix_power(Z, 4) @ np.linalg.matrix_power(Y, 3)
print(f"a) Is YZ - ZY skew-symmetric? {is_skew_symmetric(M_a)}")
```

python Code

```
# b) X^44 + Y^44
M_b = np.linalg.matrix_power(X, 44) + np.linalg.matrix_power(Y,
    44)
print(f"b) Is X + Y skew-symmetric? {is_skew_symmetric(M_b)}")

# c) X^4 * Z^3 - Z^3 * X^4
M_c = np.linalg.matrix_power(X, 4) @ np.linalg.matrix_power(Z, 3)
     - \
      np.linalg.matrix_power(Z, 3) @ np.linalg.matrix_power(X, 4)
```

python Code

```
print(f"c) Is XZ - ZX skew-symmetric? {is_skew_symmetric(M_c)}")

# d) X^23 + Y^23
M_d = np.linalg.matrix_power(X, 23) + np.linalg.matrix_power(Y,
    23)
print(f"d) Is X + Y skew-symmetric? {is_skew_symmetric(M_d)}")
print("-" * 35)
```

C Code

```c
#include <stdio.h>
#include <math.h>
#include <string.h>
#define N 3
void print_matrix(const char* name, double mat[N][N]) {
    printf("Matrix %s:\n", name);
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            printf("%9.2f ", mat[i][j]);
        }
        printf("\n");
    }
    printf("\n");
}
```

C Code

```c
// Multiplies two 3x3 matrices: C = A * B
void multiply_matrices(double a[N][N], double b[N][N], double c[N
    ][N]) {
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            c[i][j] = 0;
            for (int k = 0; k < N; k++) {
                c[i][j] += a[i][k] * b[k][j];
            }
        }
    }
}
```

C Code

```c
// Calculates matrix power: result = mat^p
void power_matrix(double mat[N][N], int p, double result[N][N]) {
    // Initialize result as identity matrix
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            result[i][j] = (i == j) ? 1.0 : 0.0;
        }
    }
    double temp[N][N];
    for (int i = 0; i < p; i++) {
        multiply_matrices(result, mat, temp);
        memcpy(result, temp, sizeof(double) * N * N);
    }
}
```

C Code

```c
// Adds two matrices: C = A + B
void add_matrices(double a[N][N], double b[N][N], double c[N][N]) {
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            c[i][j] = a[i][j] + b[i][j];
        }
    }
}
```

C Code

```
// Subtracts two matrices: C = A - B
void subtract_matrices(double a[N][N], double b[N][N], double c[N
    ][N]) {
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            c[i][j] = a[i][j] - b[i][j];
        }
    }
}
```

C Code

```c
// Checks if a matrix is skew-symmetric (M^T = -M)
int is_skew_symmetric(double mat[N][N]) {
    const double epsilon = 1e-9; // Tolerance for float
        comparison
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            // Check if M[i][j] is approximately -M[j][i]
            if (fabs(mat[i][j] + mat[j][i]) > epsilon) {
                return 0; // Not skew-symmetric
            }
        }
    }
```

C Code

```c
    // Check if diagonal elements are zero
    for (int i = 0; i < N; i++) {
        if (fabs(mat[i][i]) > epsilon) {
            return 0; // Not skew-symmetric
        }
    }
    return 1; // It is skew-symmetric
}
```

C Code

```c
int main() {
    // Define arbitrary non-zero matrices as per the problem
        statement
    // X and Y are skew-symmetric (M[i][j] = -M[j][i])
    double X[N][N] = {
        {0.0, 1.0, 2.0},
        {-1.0, 0.0, 3.0},
        {-2.0, -3.0, 0.0}
    };
    double Y[N][N] = {
        {0.0, -2.0, 4.0},
        {2.0, 0.0, -5.0},
        {-4.0, 5.0, 0.0}
    };
```

C Code

```c
// Z is symmetric (M[i][j] = M[j][i])
double Z[N][N] = {
    {1.0, 2.0, 3.0},
    {2.0, 4.0, 5.0},
    {3.0, 5.0, 6.0}
};
// Temporary matrices for calculations
double term1[N][N], term2[N][N], result[N][N];
// --- Option (a): Y^3 * Z^4 - Z^4 * Y^3 ---
power_matrix(Y, 3, term1);
power_matrix(Z, 4, term2);
double Y3Z4[N][N];
multiply_matrices(term1, term2, Y3Z4);
```

C Code

```c
    power_matrix(Z, 4, term1);
    power_matrix(Y, 3, term2);
    double Z4Y3[N][N];
    multiply_matrices(term1, term2, Z4Y3);

    subtract_matrices(Y3Z4, Z4Y3, result);
    printf("a) Y^3*Z^4 - Z^4*Y^3 is skew-symmetric? --> %s\n",
        is_skew_symmetric(result) ? "Yes" : "No");
```

C Code

```c
// --- Option (b): X^44 + Y^44 ---
power_matrix(X, 44, term1);
power_matrix(Y, 44, term2);
add_matrices(term1, term2, result);
printf("b) X^44 + Y^44 is skew-symmetric? --> %s\n",
    is_skew_symmetric(result) ? "Yes" : "No");

// --- Option (c): X^4 * Z^3 - Z^3 * X^4 ---
power_matrix(X, 4, term1);
power_matrix(Z, 3, term2);
double X4Z3[N][N];
multiply_matrices(term1, term2, X4Z3);
```

C Code

```c
    power_matrix(Z, 3, term1);
    power_matrix(X, 4, term2);
    double Z3X4[N][N];
    multiply_matrices(term1, term2, Z3X4);

    subtract_matrices(X4Z3, Z3X4, result);
    printf("c) X^4*Z^3 - Z^3*X^4 is skew-symmetric? --> %s\n",
        is_skew_symmetric(result) ? "Yes" : "No");
```

C Code

```c
    // --- Option (d): X^23 + Y^23 ---
    power_matrix(X, 23, term1);
    power_matrix(Y, 23, term2);
    add_matrices(term1, term2, result);
    printf("d) X^23 + Y^23 is skew-symmetric? --> %s\n",
        is_skew_symmetric(result) ? "Yes" : "No");
    printf("\n
        ----------------------------------------------\n"
        )
    printf("Conclusion: The correct options are (c) and (d).\n");

    return 0;
}
```

Python and C Code

```
from ctypes import c_double
# Create a 3x3 matrix using ctypes
def create_matrix():
    return [[c_double(0.0) for _ in range(3)] for _ in range(3)]

# Example usage
mat = create_matrix()
mat[0][0] = c_double(1.0)
print(mat[0][0].value)
```