2.4.22

INDHIRESH S - EE25BTECH11027

August, 2025

Question

Find the equation of a plane which bisects perpendicularly the line joining the points $A(2, 3, 4)$ and $B(4, 5, 8)$ at right angles.

Equation

Let,

$$\mathbf{A} = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix} \quad \textit{and} \quad \mathbf{B} = \begin{pmatrix} 4 \\ 5 \\ 8 \end{pmatrix} \tag{1}$$

Theoretical Solution

Given that the plane is a perpendicular bisector to the line joining points A and B. Since it is a perpendicular bisector to the line joining points A and B , the midpoint of the line joining points A and B lies on the plane.
Let the midpoint of points **A** and **B** be **X**. Then

$$||\mathbf{X} - \mathbf{A}||^2 = ||\mathbf{X} - \mathbf{B}||^2 \tag{2}$$

$$||\mathbf{X}||^2 + ||\mathbf{A}||^2 - 2||\mathbf{X}|| \; ||\mathbf{A}|| = ||\mathbf{X}||^2 + ||\mathbf{B}||^2 - 2||\mathbf{X}|| \; ||\mathbf{B}|| \tag{3}$$

$$||\mathbf{A}||^2 - 2||\mathbf{X}|| \; ||\mathbf{A}|| = ||\mathbf{B}||^2 - 2||\mathbf{X}|| \; ||\mathbf{B}|| \tag{4}$$

$$||\mathbf{A}||^2 - ||\mathbf{B}||^2 = 2\mathbf{X}^{\mathbf{T}}\mathbf{A} - 2\mathbf{X}^{\mathbf{T}}\mathbf{B} \tag{5}$$

$$||\mathbf{A}||^2 - ||\mathbf{B}||^2 = 2\mathbf{X}^{\mathbf{T}}(\mathbf{A} - \mathbf{B}) \tag{6}$$

which can be written as :

$$||\mathbf{A}||^2 - ||\mathbf{B}||^2 = 2(\mathbf{A} - \mathbf{B})^T\mathbf{X} \tag{7}$$

$$(\mathbf{A}-\mathbf{B})^T\mathbf{X} = \frac{||\mathbf{A}||^2 - ||\mathbf{B}||^2}{2} \tag{8}$$

Now,

$$\mathbf{A}-\mathbf{B} = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix} - \begin{pmatrix} 4 \\ 5 \\ 8 \end{pmatrix} \tag{9}$$

$$\mathbf{A}-\mathbf{B} = \begin{pmatrix} -2 \\ -2 \\ -4 \end{pmatrix} \tag{10}$$

And

$$||\mathbf{A}|| = \sqrt{\mathbf{A^T A}} \tag{11}$$

$$||\mathbf{A}|| = \sqrt{2(2)+3(3)+4(4)} \tag{12}$$

Theoretical Solution

$$||\mathbf{A}|| = \sqrt{29} \tag{13}$$

$$||\mathbf{B}|| = \sqrt{\mathbf{B^T B}} \tag{14}$$

$$||\mathbf{B}|| = \sqrt{4(4) + 5(5) + 8(8)} \tag{15}$$

$$||\mathbf{B}|| = \sqrt{105} \tag{16}$$

Now substituting the respective value in Eq.8:

$$\begin{pmatrix} -2 \\ -2 \\ -4 \end{pmatrix}^T \mathbf{X} = \frac{29 - 105}{2} \tag{17}$$

Theoretical Solution

$$\begin{pmatrix}1\\1\\2\end{pmatrix}^T \mathbf{X} = 19 \tag{18}$$

Hence the equation of the plane is given by:

$$\begin{pmatrix}1\\1\\2\end{pmatrix}^T \mathbf{X} = 19 \tag{19}$$

C Code - Midpoint formula

```c
#include <stdio.h>

void midpoint(double A[3], double B[3], double M[3]) {
    M[0] = (A[0] + B[0]) / 2.0;
    M[1] = (A[1] + B[1]) / 2.0;
    M[2] = (A[2] + B[2]) / 2.0;
}

void normal(double A[3], double B[3], double N[3]) {
    N[0] = B[0] - A[0];
    N[1] = B[1] - A[1];
    N[2] = B[2] - A[2];
}

// Compute plane constant: d = -(N  M)
double plane_constant(double N[3], double M[3]) {
    return -(N[0]*M[0] + N[1]*M[1] + N[2]*M[2]);
}
```

Python Code

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

# Load compiled C library
lib = ctypes.CDLL(./libgeometry.so) # use geometry.dll on Windows

# Define argument and return types
lib.midpoint.argtypes = [np.ctypeslib.ndpointer(dtype=np.double,
    ndim=1, shape=3),
                         np.ctypeslib.ndpointer(dtype=np.double,
                             ndim=1, shape=3),
                         np.ctypeslib.ndpointer(dtype=np.double,
                             ndim=1, shape=3)]
```

Python Code

```
lib.normal.argtypes = [np.ctypeslib.ndpointer(dtype=np.double,
    ndim=1, shape=3),
                       np.ctypeslib.ndpointer(dtype=np.double, ndim
                           =1, shape=3),
                       np.ctypeslib.ndpointer(dtype=np.double, ndim
                           =1, shape=3)]

lib.plane_constant.argtypes = [np.ctypeslib.ndpointer(dtype=np.
    double, ndim=1, shape=3),
                               np.ctypeslib.ndpointer(dtype=np.
                                   double, ndim=1, shape=3)]
lib.plane_constant.restype = ctypes.c_double

# Input points
A = np.array([2.0, 3.0, 4.0], dtype=np.double)
B = np.array([4.0, 5.0, 8.0], dtype=np.double)
M = np.zeros(3, dtype=np.double)
N = np.zeros(3, dtype=np.double)
```

Python Code

```python
# Call C functions
lib.midpoint(A, B, M)
lib.normal(A, B, N)
d = lib.plane_constant(N, M)

# Plane equation function
def plane_z(x, y):
    return (-N[0] * x - N[1] * y - d) / N[2]

# Create small plane patch around M
span = 1.5
xx, yy = np.meshgrid(
    np.linspace(M[0] - span, M[0] + span, 10),
    np.linspace(M[1] - span, M[1] + span, 10)
)
zz = plane_z(xx, yy)
```

Python Code

```
# --- Plotting ---
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Mark and label points
ax.scatter(*A, color='red', s=100)
ax.text(A[0], A[1], A[2], A(2,3,4), color='red')

ax.scatter(*B, color='green', s=100)
ax.text(B[0], B[1], B[2], B(4,5,8), color='green')

ax.scatter(*M, color='purple', s=200, marker='*')
ax.text(M[0], M[1], M[2], M(3,4,6), color='purple')

# Line AB
ax.plot([A[0], B[0]], [A[1], B[1]], [A[2], B[2]],
        color='blue', label=Line AB)
```

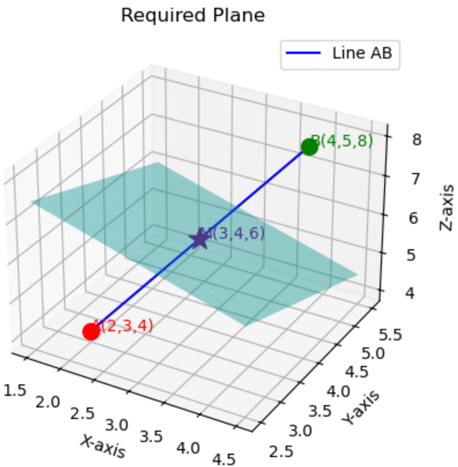
Python Code

```


# Plane patch
ax.plot_surface(xx, yy, zz, alpha=0.4, color='cyan')

# Labels and title
ax.set_xlabel(X-axis)
ax.set_ylabel(Y-axis)
ax.set_zlabel(Z-axis)
ax.set_title(Required Plane)
ax.legend()
plt.savefig(/media/indhiresh-s/New Volume/Matrix/ee1030-2025/
    ee25btech11027/MATGEO/2.4.22/figs/figure1.png)
plt.show()
```

Plot



Required Plane
Line AB
B(4,5,8)
M(3,4,6)
A(2,3,4)
X-axis
Y-axis
Z-axis
1.5
2.0
2.5
3.0
3.5
4.0
4.5
2.5
3.0
3.5
4.0
4.5
5.0
5.5
4
5
6
7
8