2.10.5

Varun-ai25btech11016

September 3, 2025

Question

A, B, C and D, are four points in a plane respectively such that $(A - D) \cdot (B - C) = (B - D) \cdot (C - A) = 0$. The point D, then, is the _____ of $\triangle ABC$.

Theoretical Solution

Consider the equation,

$$\left(A - D\right)^T \left(B - C\right) = 0 \tag{1}$$

This implies line joining A and D is perpendicular to line joining B and C
Consider the equation,

$$\left(B - D\right)^T \left(C - A\right) = 0 \tag{2}$$

This implies line joining B and D is perpendicular to line joining A and C
In $\triangle ABC$,
side BC is perpendicular to AD
side AC is perpendicular to BD

Conclusion

Therefore,
D must be Orthocenter of $\triangle ABC$
Since
The line joining vertex and orthocenter is perpendicular to opposite side

Verification by example:

Let us take the points

$\mathbf{A} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 5 \\ 1 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \mathbf{D} = \begin{pmatrix} 3 \\ \frac{7}{3} \end{pmatrix}.$

Checking the First condition:

$$(\mathbf{A} - \mathbf{D})^T (\mathbf{B} - \mathbf{C}) = 0 \tag{3}$$

$$L.H.S = \left(\begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 3 \\ \frac{7}{3} \end{pmatrix} \right)^T \left(\begin{pmatrix} 5 \\ 1 \end{pmatrix} - \begin{pmatrix} 3 \\ 4 \end{pmatrix} \right) \tag{4}$$

$$= \begin{pmatrix} -2 \\ \frac{-4}{3} \end{pmatrix}^T \begin{pmatrix} 2 \\ -3 \end{pmatrix} \tag{5}$$

$$= 0 \tag{6}$$

$$= R.H.S \tag{7}$$

$$L.H.S = R.H.S \tag{8}$$

Verification by example:

Checking the Second condition:

$$(\mathbf{B}-\mathbf{D})^T(\mathbf{C}-\mathbf{A}) = 0 \tag{9}$$

$$L.H.S = \left(\begin{pmatrix}5\\1\end{pmatrix} - \begin{pmatrix}3\\\frac{7}{3}\end{pmatrix}\right)^T \left(\begin{pmatrix}3\\4\end{pmatrix} - \begin{pmatrix}1\\1\end{pmatrix}\right) \tag{10}$$

$$= \begin{pmatrix}2\\\frac{-4}{3}\end{pmatrix}^T \begin{pmatrix}2\\3\end{pmatrix}$$

$$= 0$$

$$= R.H.S$$

$$L.H.S = R.H.S \tag{11}$$

Let's take two points F and E which are foot of perpendiculars of altitudes drawn from vertices A and B respectively.

Verification by example:

1.The normal vector of $\mathbf{F} - \mathbf{A}$ is

$$\mathbf{n} = \begin{pmatrix} 2 \\ -3 \end{pmatrix} \tag{12}$$

The equation of the altitude from A (i.e AF) is

$$\mathbf{n}^T(\mathbf{x} - \mathbf{A}) = 0 \tag{13}$$

$$\begin{pmatrix} 2 \\ -3 \end{pmatrix}^T (\mathbf{x} - \begin{pmatrix} 1 \\ 1 \end{pmatrix}) = 0 \tag{14}$$

$$\begin{pmatrix} 2 & -3 \end{pmatrix} (\mathbf{x} - \begin{pmatrix} 1 \\ 1 \end{pmatrix}) = 0 \tag{15}$$

$$\begin{pmatrix} 2 & -3 \end{pmatrix} (\mathbf{x}) = -1 \tag{16}$$

$$\tag{17}$$

Verification by example:

2. The normal vector of $\mathbf{E} - \mathbf{B}$ is

$$\mathbf{n} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \tag{18}$$

The equation of the altitude from B (i.e BE) is

$$\mathbf{n}^T(\mathbf{x} - \mathbf{B}) = 0 \tag{19}$$

$$\begin{pmatrix} 2 \\ 3 \end{pmatrix}^T (\mathbf{x} - \begin{pmatrix} 5 \\ 1 \end{pmatrix}) = 0 \tag{20}$$

$$\begin{pmatrix} 2 & 3 \end{pmatrix} (\mathbf{x} - \begin{pmatrix} 5 \\ 1 \end{pmatrix}) = 0 \tag{21}$$

$$\begin{pmatrix} 2 & 3 \end{pmatrix} (\mathbf{x}) = 13 \tag{22}$$

$$\tag{23}$$

Verification by example:

The intersection point of altitudes **orthocenter:H** can be obtained by solving the above two equations

$$\begin{pmatrix} 2 & -3 \\ 2 & 3 \end{pmatrix} \mathbf{x} = \begin{pmatrix} -1 \\ 13 \end{pmatrix} \tag{24}$$

$$\begin{pmatrix} 2 & -3 & -1 \\ 2 & 3 & 13 \end{pmatrix} \xrightarrow{R_2 \leftarrow R_2 - R_1} \begin{pmatrix} 2 & -3 & -1 \\ 0 & 6 & 14 \end{pmatrix} \tag{25}$$

$$\xrightarrow{R_2 \leftarrow \frac{1}{6} R_2} \begin{pmatrix} 2 & -3 & -1 \\ 0 & 1 & \frac{7}{3} \end{pmatrix} \tag{26}$$

$$\xrightarrow{R_1 \leftarrow R_1 + 3R_2} \begin{pmatrix} 2 & 0 & 6 \\ 0 & 1 & \frac{7}{3} \end{pmatrix} \tag{27}$$

$$\xrightarrow{R_1 \leftarrow \frac{1}{2} R_1} \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & \frac{7}{3} \end{pmatrix} \tag{28}$$

Verification by example:

which gives,

$$H = \begin{pmatrix} 3 \\ \frac{7}{3} \end{pmatrix} \tag{29}$$

Therefore,
The D we have taken coincides with the orthocenter H of the given triangle

C code

```c
// orthocenter.c
#include <stdio.h>

// Function to compute orthocenter of triangle ABC
// A, B, C are arrays of length 2: [x, y]
// D is output array of length 2: [x, y]
void orthocenter(double *A, double *B, double *C, double *D) {
    // Slopes of sides
    double m_BC = (C[1] - B[1]) / (C[0] - B[0]);
    double m_AC = (C[1] - A[1]) / (C[0] - A[0]);
```

C code

```c
    // Slopes of altitudes (negative reciprocal)
    double m_alt_A = -1.0 / m_BC;
    double m_alt_B = -1.0 / m_AC;

    // Equation of altitude from A: y - A_y = m_alt_A(x - A_x)
    // Equation of altitude from B: y - B_y = m_alt_B(x - B_x)

    double x_num = (m_alt_A*A[0] - m_alt_B*B[0] + B[1] - A[1]);
    double x_den = (m_alt_A - m_alt_B);
    double x = x_num / x_den;
    double y = m_alt_A*(x - A[0]) + A[1];

    D[0] = x;
    D[1] = y;
}
```

C plus Python code

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

# Load shared library (make sure libortho.so is in the same
    folder)
lib = ctypes.CDLL(./libortho.so)

# Define C function signature
lib.orthocenter.argtypes = [ctypes.POINTER(ctypes.c_double),
                            ctypes.POINTER(ctypes.c_double),
                            ctypes.POINTER(ctypes.c_double),
                            ctypes.POINTER(ctypes.c_double)]
```

C plus Python code

```
# Define triangle vertices
A = np.array([1.0, 1.0], dtype=np.double)
B = np.array([5.0, 1.0], dtype=np.double)
C = np.array([3.0, 4.0], dtype=np.double)
D = np.zeros(2, dtype=np.double)

# Call C function
lib.orthocenter(A.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
    ,
              B.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
              C.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
              D.ctypes.data_as(ctypes.POINTER(ctypes.c_double)))

print(Orthocenter D =, D)

# ---- Plotting ----
plt.figure(figsize=(6,6))
```

C plus Python code

```
# Triangle
plt.plot([A[0],B[0]],[A[1],B[1]],'b')
plt.plot([B[0],C[0]],[B[1],C[1]],'b')
plt.plot([C[0],A[0]],[C[1],A[1]],'b')

# Lines for perpendicularity check
plt.plot([A[0], D[0]], [A[1], D[1]], 'g--', label=AD)
plt.plot([B[0], C[0]], [B[1], C[1]], 'r--', label=BC)
plt.plot([B[0], D[0]], [B[1], D[1]], 'g--', label=BD)
plt.plot([A[0], C[0]], [A[1], C[1]], 'r--', label=AC)

# Points
plt.scatter(*A, color='red')
plt.scatter(*B, color='red')
plt.scatter(*C, color='red')
plt.scatter(*D, color='purple')
```

```
# Labels
plt.text(A[0]+0.1, A[1], 'A')
plt.text(B[0]+0.1, B[1], 'B')
plt.text(C[0]+0.1, C[1], 'C')
plt.text(D[0]+0.1, D[1], 'D (Orthocenter)')

plt.legend()
plt.gca().set_aspect('equal', adjustable='box')
plt.grid(True)
plt.savefig(/sdcard/Matrix/ee1030-2025/ai25btech11016/Matgeo
/2.10.5/figs/2.10.5.png)
plt.show()
```

Python

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

# Function to find line coefficients Ax + By = C given two points
def line_coeffs(p1, p2):
    A = p2[1] - p1[1]
    B = p1[0] - p2[0]
    C = A*p1[0] + B*p1[1]
    return A, B, C
```

Python

```
# Function to find intersection of two lines (given in Ax+By=C
    form)
def intersection(L1, L2):
    A1, B1, C1 = L1
    A2, B2, C2 = L2
    det = A1*B2 - A2*B1
    if det == 0:
        raise ValueError(Lines are parallel, no intersection.)
    x = (C1*B2 - C2*B1) / det
    y = (A1*C2 - A2*C1) / det
    return np.array([x, y])

# Define triangle vertices
A = np.array([1, 1])
B = np.array([5, 1])
C = np.array([3, 4])
```

Python

```
# Slopes of sides
L_BC = line_coeffs(B, C)
L_AC = line_coeffs(A, C)

# Altitude from A (perpendicular to BC, passes through A)
A1, B1, _ = L_BC
L_alt_A = (-B1, A1, -B1*A[0] + A1*A[1])

# Altitude from B (perpendicular to AC, passes through B)
A2, B2, _ = L_AC
L_alt_B = (-B2, A2, -B2*B[0] + A2*B[1])

# Orthocenter (D)
D = intersection(L_alt_A, L_alt_B)
```

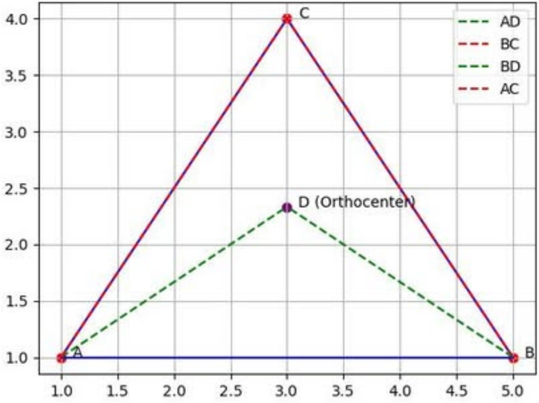
Python

```python
# Plotting
plt.figure(figsize=(6,6))
# Triangle
plt.plot([A[0],B[0]],[A[1],B[1]],'b')
plt.plot([B[0],C[0]],[B[1],C[1]],'b')
plt.plot([C[0],A[0]],[C[1],A[1]],'b')

# Lines showing perpendicularity
plt.plot([A[0], D[0]], [A[1], D[1]], 'g--', label=AD)
plt.plot([B[0], C[0]], [B[1], C[1]], 'r--', label=BC)
plt.plot([B[0], D[0]], [B[1], D[1]], 'g--', label=BD)
plt.plot([A[0], C[0]], [A[1], C[1]], 'r--', label=AC)

# Points
plt.scatter(*A, color='red')
plt.scatter(*B, color='red')
plt.scatter(*C, color='red')
plt.scatter(*D, color='purple')
```

Python

```
# Labels
plt.text(A[0]+0.1, A[1], 'A')
plt.text(B[0]+0.1, B[1], 'B')
plt.text(C[0]+0.1, C[1], 'C')
plt.text(D[0]+0.1, D[1], 'D (Orthocenter)')

plt.legend()
plt.gca().set_aspect('equal', adjustable='box')
plt.grid(True)
plt.savefig(/sdcard/Matrix/ee1030-2025/ai25btech11016/Matgeo
    /2.10.5/figs/2.10.5.png)
plt.show()
```

Plot



AD
BC
BD
AC
C
D (Orthocenter)
A
B
4.0
3.5
3.0
2.5
2.0
1.5
1.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0