1.9.30

Nipun Dasari - EE25BTECH11042

January 9, 2025

Question

If the distances of $\mathbf{P} = (x, y)$ from $\mathbf{A} = (5, 1)$ and $\mathbf{B} = (1, 5)$ are equal, then prove that 3x = 2y.

Theoretical Solution

Consider the matrices A, B and P as follows:

$$\mathbf{A} = \begin{pmatrix} 5 \\ 1 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} -1 \\ 5 \end{pmatrix}, \mathbf{P} = \begin{pmatrix} x \\ y \end{pmatrix}$$

The condition for distances from **B** to **P** and **A** to **P** to be equal is

$$\|\mathbf{P} - \mathbf{A}\| = \|\mathbf{P} - \mathbf{B}\| \equiv \|\mathbf{P} - \mathbf{A}\|^2 = \|\mathbf{P} - \mathbf{B}\|^2$$

Using inner products:

$$(\mathbf{P} - \mathbf{A})^T (\mathbf{P} - \mathbf{A}) = (\mathbf{P} - \mathbf{B})^T (\mathbf{P} - \mathbf{B})$$

Expanding on both sides:

$$\mathbf{P}\mathbf{P}^T - 2\mathbf{A}^T\mathbf{P} + \mathbf{A}^T\mathbf{A} = \mathbf{P}\mathbf{P}^T - 2\mathbf{B}^T\mathbf{P} + \mathbf{B}^T\mathbf{B}$$

On simplification:

$$\left(-2\mathbf{A}^T + 2\mathbf{B}^T\right)\mathbf{P} = \mathbf{B}^T\mathbf{B} - \mathbf{A}^T\mathbf{A}$$

Theoretical Solution

LHS constant matrix:

$$2\left(\mathbf{B}-\mathbf{A}\right)^{T} = 2\begin{pmatrix} -1-5 \\ 5-1 \end{pmatrix} = \begin{pmatrix} -12 & 8 \end{pmatrix}$$

RHS constant matrix:

$$\mathbf{B}^{T}\mathbf{B}-\mathbf{A}^{T}\mathbf{A} = \left((-1)^{2}+5^{2}\right)-\left(1^{2}+5^{2}\right) = 0$$

From the above:

$$\begin{pmatrix} -12 & 8 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = 0 \implies -12x+8y=0 \implies 3x=2y$$

C Code- equidistant check function

```c
#include <math.h>

// Returns 1 if 3x=2y within tolerance, else 0
int equidist_check(double x, double y, double tol) {
        double val = 3.0*x - 2.0*y;
        return fabs(val) <= tol ? 1 : 0;
}
```

Python Code using shared output

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

# Load the shared library
lib = ctypes.CDLL(os.path.abspath(./libequidist.so))
lib.equidist_check.argtypes = [ctypes.c_double, ctypes.c_double, ctypes.c_double]
lib.equidist_check.restype = ctypes.c_int

def is_on_bisector(x, y, tol=1e-9):
Check if point (x,y) satisfies 3x = 2y using C library
return bool(lib.equidist_check(x, y, tol))
```

Python Code using shared output

```
# Points A, B, midpoint
A = np.array([5, 1])
B = np.array([-1, 5])
midpoint = (A + B) / 2

# Equation: 3x=2y -> y=(3/2)x
x_vals = np.linspace(-2, 6, 200)
y_vals = (3/2) * x_vals

# Example points to check
points_to_check = [A, B, midpoint]

# Print verification results using C library
for p in points_to_check:
result = is_on_bisector(p[0], p[1])
print(fPoint {p} on bisector? {result})
```

Python Code using shared output

```python
# Plotting
plt.figure(figsize=(6, 6))
plt.scatter(A[0], A[1], color='red', label='A(5,1)')
plt.scatter(B[0], B[1], color='blue', label='B(-1,5)')
plt.scatter(midpoint[0], midpoint[1], color='black', label='Midpoint')

# Perpendicular bisector line
plt.plot(x_vals, y_vals, 'g--', label='3x = 2y (Perp. bisector)')

# Mark example points
for p in points_to_check:
plt.scatter(p[0], p[1], label=fPoint {p})
```

Python Code using shared output

```
# Decorations
plt.title(Perpendicular Bisector of A and B: 3x=2y)
plt.xlabel(x)
plt.ylabel(y)
plt.legend()
plt.grid(True)
plt.axis('equal')
plt.show()
```

Plot by python using shared output from c