1.8.22: Equidistant Points Problem

Aditya Mishra - ee25btech11005

September 9, 2025

Question

Find all points that are equidistant from

$$\mathbf{A} = \begin{pmatrix} -5 \\ 4 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} -1 \\ 6 \end{pmatrix}.$$

How many such points exist?

Given Information

Points as column vectors:

$$\mathbf{A} = \begin{pmatrix} -5 \\ 4 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} -1 \\ 6 \end{pmatrix}.$$

Desired point:

$$\mathbf{O} = \begin{pmatrix} x \\ y \end{pmatrix}.$$

Equidistant Condition

Equidistant means:

$$\|\mathbf{O} - \mathbf{A}\| = \|\mathbf{O} - \mathbf{B}\| .$$

Squaring both sides:

$$\|\mathbf{O} - \mathbf{A}\|^2 = \|\mathbf{O} - \mathbf{B}\|^2 .$$

Using vector dot product:

$$(\mathbf{O} - \mathbf{A})^\top (\mathbf{O} - \mathbf{A}) = (\mathbf{O} - \mathbf{B})^\top (\mathbf{O} - \mathbf{B}).$$

Simplify the Equation

Expand both sides:

$$\mathbf{O}^\top \mathbf{O} - 2\mathbf{A}^\top \mathbf{O} + \mathbf{A}^\top \mathbf{A} = \mathbf{O}^\top \mathbf{O} - 2\mathbf{B}^\top \mathbf{O} + \mathbf{B}^\top \mathbf{B}.$$

Simplify by cancelling $\mathbf{O}^\top \mathbf{O}$:

$$-2\mathbf{A}^\top \mathbf{O} + \mathbf{A}^\top \mathbf{A} = -2\mathbf{B}^\top \mathbf{O} + \mathbf{B}^\top \mathbf{B}.$$

Rearranged:

$$2(\mathbf{B} - \mathbf{A})^\top \mathbf{O} = \mathbf{B}^\top \mathbf{B} - \mathbf{A}^\top \mathbf{A}.$$

Final Equation

$$(\mathbf{B}-\mathbf{A})^\top \mathbf{O} = \frac{\mathbf{B}^\top \mathbf{B} - \mathbf{A}^\top \mathbf{A}}{2}.$$

Put terms explicitly:

$$\begin{pmatrix} 4 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{37-41}{2} = -2.$$

That gives a line equation:

$$4x + 2y = -2 \implies 2x + y = -1.$$

Number of Solutions

The set of points equidistant from **A** and **B** lies on the line:

$$2x + y = -1.$$

There are infinitely many such points.

C Code: Equidistant Line Calculation

```c
#include <stdio.h>

void equidistant_line(double ax, double ay, double bx, double by,
double *res) {
    double a = bx - ax;
    double b = by - ay;
    double normA_sq = ax*ax + ay*ay;
    double normB_sq = bx*bx + by*by;
    double c = (normB_sq - normA_sq) / 2.0;

    res[0] = a;
    res[1] = b;
    res[2] = c;
}
```

Python ctypes

```
import ctypes

lib = ctypes.CDLL('./libequidistant.so')

lib.equidistant_line.argtypes = [
    ctypes.c_double, ctypes.c_double,
    ctypes.c_double, ctypes.c_double,
    ctypes.POINTER(ctypes.c_double),  # Output array for coeffs
]

lib.equidistant_line.restype = None

res = (ctypes.c_double * 3)()  # To hold a,b,c in ax + by = c

lib.equidistant_line(-5, 4, -1, 6, res)

print(f"Line: {res[0]} x + {res[1]} y = {res[2]}")
```

Python Code: Plot Equidistant Line and Points

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

A = np.array([-5, 4])
B = np.array([-1, 6])

a = B[0] - A[0]
b = B[1] - A[1]

c = (np.dot(B, B) - np.dot(A, A)) / 2
x = np.linspace(-10, 5, 400)
y = (c - a * x) / b
```

Python Code: Plot Equidistant Line and Points

```
plt.scatter(*A, color='red', label='A(-5,4)')
plt.scatter(*B, color='green', label='B(-1,6)')

plt.plot(x, y, color='blue', label='Equidistant line')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')

plt.title('Equidistant Points: Line and Given Points')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.xlim(-10, 20)
plt.ylim(-10, 20)
plt.savefig('equidistant_plot.png')
plt.show()
```

Plot

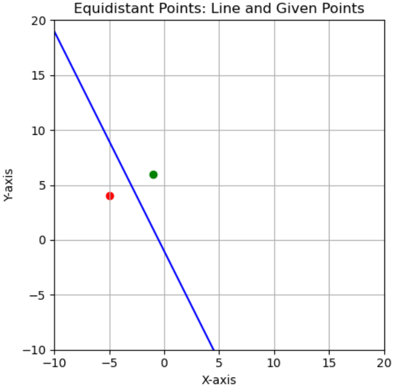

Figure: Points A, B and equidistant line $2x + y = -1$