4.13.6

EE25BTECH11001 - Aarush Dilawri

October 10, 2025

Problem Statement

Question:
Given the points $\mathbf{A}\left(0, 4\right)$ and $\mathbf{B}\left(0, -4\right)$, the equation of the locus of the point $\mathbf{p}\left(x, y\right)$, such that $\left(AP - BP\right)^2 = 6^2$

Step 1: Define Vectors

$$\mathbf{A},\ \mathbf{B},\ \mathbf{P} \in \mathbb{R}^n \tag{1}$$

Let the given scalar be $\delta \geq 0$

$$(r_1 - r_2)^2 = \delta^2, \tag{2}$$

where $r_1 = \|\mathbf{P} - \mathbf{A}\|$ and $r_2 = \|\mathbf{P} - \mathbf{B}\|$.

Step 2: Taking Square root

Taking square root,

$$|r_1 - r_2| = \pm\delta \tag{3}$$

Let's define

$$D = s\,\delta \tag{4}$$

where $s \in \{+1, -1\}$ such that

$$r_1 - r_2 = s\delta = D \tag{5}$$

Step 3: Difference of Squares

Let's find $r_1^2 - r_2^2$

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

$$= -2\,\mathbf{P}^\top\mathbf{u} + \mathbf{A}^\top\mathbf{A} - \mathbf{B}^\top\mathbf{B} \;=\; -2\,\mathbf{P}^\top\mathbf{u} + \alpha. \tag{7}$$

where

$$\mathbf{u} = \mathbf{A} - \mathbf{B} \quad \text{and} \quad \alpha \;=\; \mathbf{A}^\top\mathbf{A} - \mathbf{B}^\top\mathbf{B}. \tag{8}$$

Step 4: Using $r_1^2 - r_2^2$ Identity

Use $(r_1 - r_2)(r_1 + r_2) = r_1^2 - r_2^2$ and $r_1 - r_2 = D$ to get

$$D\left(r_1 + r_2\right) = -2\,\mathbf{P}^\top \mathbf{u} + \alpha \quad \Longrightarrow \quad r_1 + r_2 \;=\; \frac{-2\,\mathbf{P}^\top \mathbf{u} + \alpha}{D}. \tag{9}$$

Hence

$$r_1 = \frac{(r_1 - r_2) + (r_1 + r_2)}{2} \;=\; \frac{D}{2} + \frac{\alpha}{2D} - \frac{\mathbf{P}^\top \mathbf{u}}{D}. \tag{10}$$

Step 5: Quadratic Form

Square this expression and equate to the explicit quadratic form for r_1^2:

$$\left(\frac{D}{2} + \frac{\alpha}{2D} - \frac{\mathbf{P}^\top \mathbf{u}}{D}\right)^2 = (\mathbf{P} - \mathbf{A})^\top (\mathbf{P} - \mathbf{A}) \;=\; \mathbf{P}^\top \mathbf{P} - 2\,\mathbf{P}^\top \mathbf{A} + \mathbf{A}^\top \mathbf{A}. \tag{11}$$

Multiply both sides by D^2 and simplify to get the general quadratic (conic) equation:

$$\mathbf{P}^\top (\mathbf{u}\mathbf{u}^\top - D^2 I)\mathbf{P} \;+\; \left(-(D^2+\alpha)\,\mathbf{u} + 2D^2\,\mathbf{A}\right)^\top \mathbf{P} \;+\; \frac{(D^2+\alpha)^2}{4} - D^2\,\mathbf{A}^\top \mathbf{A} \tag{12}$$

Step 6: Substitute Values

Substitute $\mathbf{A} = \begin{pmatrix} 0 \\ 4 \end{pmatrix}$, $\mathbf{B} = \begin{pmatrix} 0 \\ -4 \end{pmatrix}$ and $\delta = 6$.

$$\mathbf{u} = \mathbf{A} - \mathbf{B} = \begin{pmatrix} 0 \\ 8 \end{pmatrix} \tag{13}$$

$$\alpha = \mathbf{A}^\top \mathbf{A} - \mathbf{B}^\top \mathbf{B} = 16 - 16 = 0 \tag{14}$$

$$D = s\,\delta = \pm 6 \quad \Rightarrow \quad D^2 = 36 \tag{15}$$

Step 7: Quadratic Matrix Equation

$$\mathbf{P}^\top(\mathbf{u}\mathbf{u}^\top - 36I)\mathbf{P} + (-D^2\mathbf{u} + 2D^2\mathbf{A})^\top\mathbf{P} + \frac{D^4}{4} - 36\mathbf{A}^\top\mathbf{A} = 0 \tag{16}$$

Compute each term:

$$\mathbf{u}\mathbf{u}^\top = \begin{pmatrix} 0 & 0 \\ 0 & 64 \end{pmatrix}, \qquad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{17}$$

$$\mathbf{u}\mathbf{u}^\top - 36I = \begin{pmatrix} -36 & 0 \\ 0 & 28 \end{pmatrix} \tag{18}$$

Step 8: Linear and Constant Terms

Linear coefficient:

$$-D^2\mathbf{u} + 2D^2\mathbf{A} = -36\begin{pmatrix}0\\8\end{pmatrix} + 72\begin{pmatrix}0\\4\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix} \tag{19}$$

Constant term:

$$\frac{D^4}{4} - 36\mathbf{A}^\top\mathbf{A} = 324 - 576 = -252 \tag{20}$$

Step 9: Locus Equation

The locus is

$$\mathbf{P}^\top \begin{pmatrix} -36 & 0 \\ 0 & 28 \end{pmatrix} \mathbf{P} - 252 = 0 \quad \implies \quad -36x^2 + 28y^2 = 252 \tag{21}$$

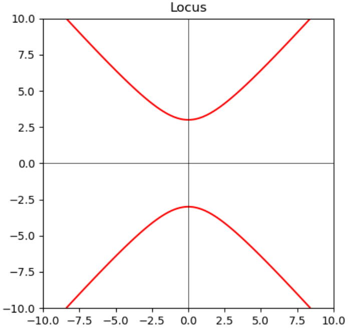
Dividing through:

$$\frac{y^2}{9} - \frac{x^2}{7} = 1 \tag{22}$$

Thus, the locus is a hyperbola centered at the origin:

$$\frac{y^2}{9} - \frac{x^2}{7} = 1 \tag{23}$$

Figure



C Code (code.c)

```
#include <stdio.h>
#include <math.h>
double inner_product(int n, double *u, double *v) {
    double sum = 0.0;
    for(int i=0; i<n; i++) {
        sum += u[i]*v[i];
    }
    return sum;
}
double locus_value(int n, double *A, double *B, double *P, double D)
    {
    // Compute u = A - B
    double u[10]; // assume dimension <= 10 for simplicity
    for(int i=0; i<n; i++) {
        u[i] = A[i] - B[i];
    }
```

C Code (code.c)

```
    double alpha = inner_product(n, A, A) - inner_product(n, B, B);
    double uuT_P[n];
    for(int i=0; i<n; i++) {
        uuT_P[i] = u[i]*inner_product(n, u, P);
    }
    double quad = inner_product(n, P, uuT_P);
    quad -= D*D * inner_product(n, P, P);
    double coeff[n];
    for(int i=0; i<n; i++) {
        coeff[i] = -(D*D+alpha)*u[i] + 2*D*D*A[i];
    }
    double lin = inner_product(n, coeff, P);
    double constant = ((D*D+alpha)*(D*D+alpha))/4.0 - D*D*
        inner_product(n, A, A);

    return quad + lin + constant;
}
```

Python Code (code.py)

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

def inner_product(u, v):
    return np.dot(u, v)

def locus_value(A, B, P, D):
    u = A - B
    alpha = inner_product(A, A) - inner_product(B, B)
    quad = (np.dot(P, u))**2 - D*D*inner_product(P, P)
    coeff = -(D*D+alpha)*u + 2*D*D*A
    lin = inner_product(coeff, P)
    constant = ((D*D+alpha)**2)/4.0 - D*D*inner_product(A, A)

    return quad + lin + constant
```

Python Code (code.py)

```
# Example: A=(0,4), B=(0,-4), delta=6
A = np.array([0.0, 4.0])
B = np.array([0.0, -4.0])
D = 6.0

x = np.linspace(-10, 10, 400)
y = np.linspace(-10, 10, 400)
X, Y = np.meshgrid(x, y)
Z = np.zeros_like(X)
```

Python Code (code.py)

```
for i in range(X.shape[0]):
    for j in range(X.shape[1]):
        P = np.array([X[i,j], Y[i,j]])
        Z[i,j] = locus_value(A, B, P, D)

plt.contour(X, Y, Z, levels=[0], colors="red")
plt.axhline(0, color="k", linewidth=0.5)
plt.axvline(0, color="k", linewidth=0.5)
plt.gca().set_aspect("equal")
plt.title("Locus using pure Python")
plt.show()
```

Python Code (nativecode.py)

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

# Load the shared library
lib = ctypes.CDLL("./code.so")

# Define function signature
lib.locus_value.argtypes = [
    ctypes.c_int, # dimension
    np.ctypeslib.ndpointer(dtype=np.double), # A
    np.ctypeslib.ndpointer(dtype=np.double), # B
    np.ctypeslib.ndpointer(dtype=np.double), # P
    ctypes.c_double # D
]
lib.locus_value.restype = ctypes.c_double
```

Python Code (nativecode.py)

```
# Example: A=(0,4), B=(0,-4), delta=6
A = np.array([0.0, 4.0], dtype=np.double)
B = np.array([0.0, -4.0], dtype=np.double)
D = 6.0

# Create grid and evaluate locus
x = np.linspace(-10, 10, 400)
y = np.linspace(-10, 10, 400)
X, Y = np.meshgrid(x, y)
Z = np.zeros_like(X)
```

Python Code (nativecode.py)

```
for i in range(X.shape[0]):
    for j in range(X.shape[1]):
        P = np.array([X[i,j], Y[i,j]], dtype=np.double)
        Z[i,j] = lib.locus_value(2, A, B, P, D)

# Plot contour Z=0 (the locus)
plt.contour(X, Y, Z, levels=[0], colors="blue")
plt.axhline(0, color="k", linewidth=0.5)
plt.axvline(0, color="k", linewidth=0.5)
plt.gca().set_aspect("equal")
plt.title("Locus using C library")
plt.show()
```