2.8.6

Vishwambhar - EE25BTECH11025

31st August, 2025

Question

Assuming that the straight lines work as the plane mirror for a point, find the image of the point $(1, 2)$ in the line $x - 3y + 4 = 0$.

Translation

Translating the system by $\mathbf{A} = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$ so that the line passes through origin:

$$L = \begin{pmatrix} 1 & -3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = -4; \mathbf{P} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \tag{1}$$

$$\mathbf{P}_{trans} = \mathbf{P} - \mathbf{A} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} - \begin{pmatrix} -4 \\ 0 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \end{pmatrix} \tag{2}$$

$$L_{trans} = \begin{pmatrix} 1 & -3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 0 \tag{3}$$

Normal Vector

Finding the normal vector:

$$\mathbf{N} = \begin{pmatrix} 1 & -3 \end{pmatrix} \tag{4}$$

Finding the unit normal vector:

$$||\mathbf{N}|| = \sqrt{1^2 + (-3)^2} = \sqrt{10} \tag{5}$$

$$\mathbf{n} = \frac{\mathbf{N}}{||\mathbf{N}||} = \frac{1}{\sqrt{10}} \begin{pmatrix} 1 \\ -3 \end{pmatrix} \tag{6}$$

Reflection Matrix

Calculating the reflection matrix R is given by the formula $R = I - 2\mathbf{n}\mathbf{n}^T$

$$R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - 2\left(\frac{1}{\sqrt{10}}\begin{pmatrix} 1 \\ -3 \end{pmatrix}\right)\left(\frac{1}{\sqrt{10}}\begin{pmatrix} 1 & -3 \end{pmatrix}\right) = \begin{pmatrix} \frac{4}{5} & \frac{3}{5} \\ \frac{3}{5} & \frac{-4}{5} \end{pmatrix} \tag{7}$$

Reflecting the given point:

$$\mathbf{P}'_{trans} = R.P_{trans} = \begin{pmatrix} \frac{26}{5} \\ \frac{7}{5} \end{pmatrix} \tag{8}$$

Conclusion

Inverting the translation:

$$\mathbf{P}' = \mathbf{P}'_{trans} + \mathbf{A} = \begin{pmatrix} \frac{6}{5} \\ \frac{7}{5} \end{pmatrix} \tag{9}$$

Thus the final image of the given point is $\mathbf{P}' = \begin{pmatrix} \frac{6}{5} \\ \frac{7}{5} \end{pmatrix}$

C Code

```c
// reflect.c
#include <math.h>

typedef struct { double x, y; } Point;
typedef struct { double a, b, c; } Line;

/* Stored values for the question */
static Point stored_point = {1.0, 2.0};
static Line stored_line = {1.0, -3.0, 4.0};

/* Accessors */
void get_point(double* x, double* y){ if(x)*x=stored_point.x; if(
    y)*y=stored_point.y; }
void get_line(double* a,double* b,double* c){ if(a)*a=stored_line
    .a; if(b)*b=stored_line.b; if(c)*c=stored_line.c; }
```

C Code

```c
/* General reflection across ax+by+c=0 */
void reflect_point_across_line(double x0, double y0,
                               double a, double b, double c,
                               double* xr, double* yr)
{
    double denom = a*a + b*b;
    double t = (a*x0 + b*y0 + c) / denom;
    if(xr) *xr = x0 - 2*a*t;
    if(yr) *yr = y0 - 2*b*t;
}

/* Convenience for stored values */
void reflect_stored(double* xr, double* yr){
    reflect_point_across_line(stored_point.x, stored_point.y,
                              stored_line.a, stored_line.b,
                                  stored_line.c,
                              xr, yr);
}
```

Python Code 1

```
# solve_reflection.py
import ctypes
from ctypes import c_double, byref

lib = ctypes.CDLL('./problem.so') # adjust path if needed

# Signatures
lib.reflect_stored.argtypes = [ctypes.POINTER(c_double), ctypes.
    POINTER(c_double)]
lib.reflect_stored.restype = None
lib.get_point.argtypes = [ctypes.POINTER(c_double), ctypes.
    POINTER(c_double)]
lib.get_point.restype = None
lib.get_line.argtypes = [ctypes.POINTER(c_double), ctypes.POINTER
    (c_double), ctypes.POINTER(c_double)]
lib.get_line.restype = None
```

Python Code 1

```
# Read stored inputs
x0 = c_double(); y0 = c_double()
a = c_double(); b = c_double(); c = c_double()
lib.get_point(byref(x0), byref(y0))
lib.get_line(byref(a), byref(b), byref(c))

# Compute reflection
xr = c_double(); yr = c_double()
lib.reflect_stored(byref(xr), byref(yr))

print(f"Point P: ({x0.value}, {y0.value})")
print(f"Line: {a.value}*x + {b.value}*y + {c.value} = 0")
print(f"Reflected image: ({xr.value}, {yr.value})") # (-1.0, 2.0)
```

Python Code 2

```
import sys
sys.path.insert(0, '/home/ganachari-vishwmabhar/Downloads/codes/
    CoordGeo')
import numpy as np
import matplotlib.pyplot as plt

from line.funcs import *
from triangle.funcs import *

# Given line: x - 3y + 4 = 0 => a=1, b=-3, c=4
a, b, c = 1, -3, 4

P = np.array([1, 2])

x1, y1 = P
den = a**2 + b**2
```

Python Code 2

```
x_img = ((b**2 - a**2)*x1 - 2*a*b*y1 - 2*a*c)/den
y_img = ((a**2 - b**2)*y1 - 2*a*b*x1 - 2*b*c)/den
P_img = np.array([x_img, y_img])
# Plot line
x_vals = np.linspace(-5, 5, 100)
y_vals = (-(a*x_vals + c))/b
plt.plot(x_vals, y_vals, 'k-', label='Mirror Line')

# Plot original point and image
plt.scatter([P[0], P_img[0]], [P[1], P_img[1]], c=['r','b'])
plt.text(P[0], P[1], 'P(1,2)', fontsize=12)
plt.text(P_img[0], P_img[1], "P'", fontsize=12)
```

Python Code 2

```
# Connect them with perpendicular
plt.plot([P[0], P_img[0]], [P[1], P_img[1]], 'g--', label='
    Perpendicular')

# Settings
plt.axis('equal')
plt.grid(True)
plt.legend()
plt.title("Reflection of Point (1,2) in Line x - 3y + 4 = 0")
plt.savefig("../figs/plot.png")
plt.show()
```

Plot

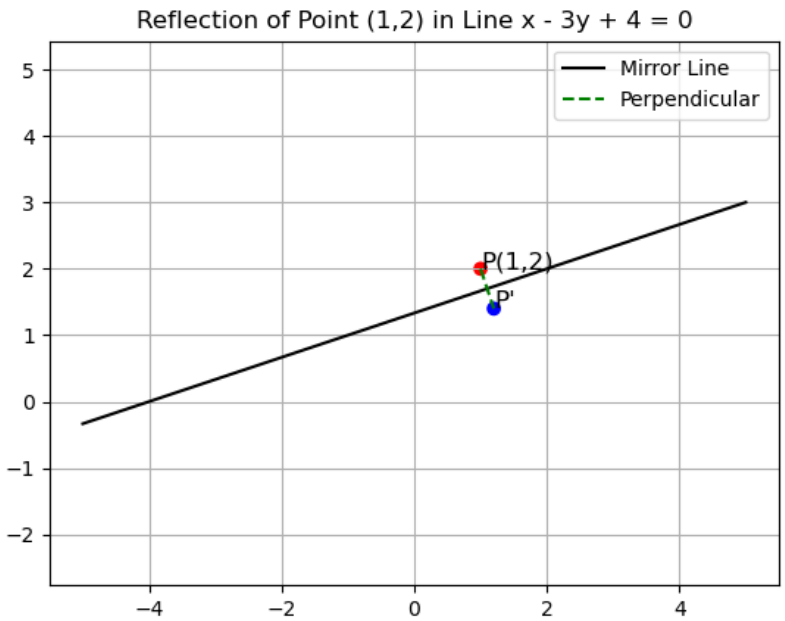

Figure: Plot of orthogonal vectors **a** and **b**.