

## 4.13.41

M Chanakya Srinivas- EE25BTECH11036

# Problem

Find the area of the parallelogram formed by the lines

$$y = mx, \quad y = mx + 1, \quad y = nx, \quad y = nx + 1.$$

# Normal Form of Lines

The general equation of a line is

$$\mathbf{n}^\top \mathbf{x} = c \quad (1)$$

where  $\mathbf{n}$  is the normal vector.

$$y - mx = 0 \quad \Rightarrow \quad \mathbf{n}_1^\top \mathbf{x} = 0, \quad (2)$$

$$y - mx - 1 = 0 \quad \Rightarrow \quad \mathbf{n}_2^\top \mathbf{x} = 1, \quad (3)$$

$$y - nx = 0 \quad \Rightarrow \quad \mathbf{n}_3^\top \mathbf{x} = 0, \quad (4)$$

$$y - nx - 1 = 0 \quad \Rightarrow \quad \mathbf{n}_4^\top \mathbf{x} = 1. \quad (5)$$

$$\mathbf{n}_1 = \mathbf{n}_2 = \begin{pmatrix} -m \\ 1 \end{pmatrix}, \quad \mathbf{n}_3 = \mathbf{n}_4 = \begin{pmatrix} -n \\ 1 \end{pmatrix}. \quad (6)$$

# Intersection of Two Lines

For two lines  $\mathbf{n}_1^\top \mathbf{x} = c_1$ ,  $\mathbf{n}_2^\top \mathbf{x} = c_2$ ,

$$\begin{pmatrix} \mathbf{n}_1^\top \\ \mathbf{n}_2^\top \end{pmatrix} \mathbf{x} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}. \quad (7)$$

Hence,

$$\mathbf{x} = \begin{pmatrix} \mathbf{n}_1^\top \\ \mathbf{n}_2^\top \end{pmatrix}^{-1} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}. \quad (8)$$

$$\mathbf{N} = \begin{pmatrix} -m & 1 \\ -n & 1 \end{pmatrix}, \quad \mathbf{N}^{-1} = \frac{1}{n-m} \begin{pmatrix} 1 & -1 \\ n & -m \end{pmatrix}. \quad (9)$$

# Vertices of Parallelogram

$$\mathbf{A} = \mathbf{N}^{-1} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (10)$$

$$\mathbf{B} = \mathbf{N}^{-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{n-m} \begin{pmatrix} 1 \\ n \end{pmatrix}, \quad (11)$$

$$\mathbf{C} = \mathbf{N}^{-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (12)$$

$$\mathbf{D} = \mathbf{N}^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{n-m} \begin{pmatrix} -1 \\ -m \end{pmatrix}. \quad (13)$$

Thus,

$$\mathbf{A} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{B} = \frac{1}{n-m} \begin{pmatrix} 1 \\ n \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \mathbf{D} = \frac{1}{n-m} \begin{pmatrix} -1 \\ -m \end{pmatrix}. \quad (14)$$

# Area of Parallelogram

Two adjacent sides:

$$\mathbf{B} - \mathbf{A} = \frac{1}{n - m} \begin{pmatrix} 1 \\ n \end{pmatrix}, \quad (15)$$

$$\mathbf{D} - \mathbf{A} = \frac{1}{n - m} \begin{pmatrix} -1 \\ -m \end{pmatrix}. \quad (16)$$

Area:

$$\text{Area} = |(\mathbf{B} - \mathbf{A}) \times (\mathbf{D} - \mathbf{A})| \quad (17)$$

$$= \left| \frac{1}{(n - m)^2} \det \begin{pmatrix} 1 & -1 \\ n & -m \end{pmatrix} \right| \quad (18)$$

$$= \frac{1}{|m - n|}. \quad (19)$$

# FINAL ANSWER

$$\text{Area of the parallelogram} = \frac{1}{|m - n|}$$

```
#include <math.h>

// Make the symbol visible in the shared object
__attribute__((visibility(default)))
double parallelogram_area(double x1, double y1, double x2, double
    y2) {
    // Cross product magnitude
    return fabs(x1 * y2 - x2 * y1);
}
```



# Python code through shared output

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

# Load the compiled shared library
lib = ctypes.CDLL('./libpara.so')
lib.parallelogram_area.argtypes = [ctypes.c_double, ctypes.c_double, ctypes.c_double, ctypes.c_double]
lib.parallelogram_area.restype = ctypes.c_double

# Slopes
m = 1
n = -1
```

# Python code through shared output

```
# Function to get intersection of two lines:  $y = m_1 x + c_1$  and  $y = m_2 x + c_2$ 
def line_intersection(m1, c1, m2, c2):
    x = (c2 - c1) / (m1 - m2)
    y = m1 * x + c1
    return x, y

# Calculate all four vertices (intersections)
P1 = line_intersection(m, 0, n, 0) #  $y=mx$  and  $y=nx$ 
P2 = line_intersection(m, 0, n, 1) #  $y=mx$  and  $y=nx+1$ 
P3 = line_intersection(m, 1, n, 1) #  $y=mx+1$  and  $y=nx+1$ 
P4 = line_intersection(m, 1, n, 0) #  $y=mx+1$  and  $y=nx$ 

# Convert to numpy arrays for vector operations
P1 = np.array(P1)
P2 = np.array(P2)
```

# Python code through shared output

```
P3 = np.array(P3)
P4 = np.array(P4)

# Compute side vectors for area calculation (two adjacent sides
    from P1)
vec1 = P2 - P1
vec2 = P4 - P1

# Call the C function for area
area = lib.parallelogram_area(vec1[0], vec1[1], vec2[0], vec2[1])

# ----- Plotting Section -----

# X range for lines plotting
x_vals = np.linspace(-2, 2, 400)
```

# Python code through shared output

```
# Lines  $y = mx$  and  $y = mx + 1$ 
y_m = m * x_vals
y_m1 = m * x_vals + 1

# Lines  $y = nx$  and  $y = nx + 1$ 
y_n = n * x_vals
y_n1 = n * x_vals + 1

plt.figure(figsize=(8, 8))
plt.plot(x_vals, y_m, label=r'$y = mx$', color='blue')
plt.plot(x_vals, y_m1, label=r'$y = mx + 1$', linestyle='--',
         color='blue')
plt.plot(x_vals, y_n, label=r'$y = nx$', color='red')
plt.plot(x_vals, y_n1, label=r'$y = nx + 1$', linestyle='--',
         color='red')
```

# Python code through shared output

```
# Parallelogram vertices for plotting (close the polygon by
    adding P1 again)
vertices_x = [P1[0], P2[0], P3[0], P4[0], P1[0]]
vertices_y = [P1[1], P2[1], P3[1], P4[1], P1[1]]

# Plot parallelogram
plt.plot(vertices_x, vertices_y, 'k-', linewidth=2, label='
    Parallelogram')
plt.fill(vertices_x, vertices_y, color='gray', alpha=0.3)

# Label vertices
for i, (xv, yv) in enumerate(zip(vertices_x[:-1], vertices_y
   [:-1]), 1):
    plt.text(xv, yv, f'P{i}', fontsize=12, ha='right', va='bottom
        ')
```

# Python code through shared output

```
# Show area on plot
plt.text(-1.5, 1.5, f'Area = {area:.4f}', fontsize=14, color='
green',
        bbox=dict(facecolor='white', alpha=0.8))

# Axis formatting
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
```

# Python code through shared output

```
plt.grid(True, linestyle='--', alpha=0.5)
plt.legend()
plt.title('Parallelogram formed by lines  $y=mx$ ,  $y=mx+1$ ,  $y=nx$ ,  $y=nx+1$ \n(Area via C shared library)')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.axis('equal')
plt.xlim(-2, 2)
plt.ylim(-2, 2)

plt.show()
```

# Only Python code

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

```
def line_intersect(n1, c1, n2, c2):
```

Find intersection point of two lines:

$n1.T * x = c1$

$n2.T * x = c2$

Inputs:

$n1, n2$ : (2,) or (2,1) normal vectors to lines

$c1, c2$ : scalars

Returns:

$p$ : (2,1) intersection point

```
N = np.column_stack((n1.flatten(), n2.flatten())) # 2x2
matrix
```

```
C = np.array([c1, c2])
```

```
p = np.linalg.solve(N, C)
```

```
return p.reshape(2,1)
```



# Only Python code

```
def line_dir_pt(direction, point, k1, k2, num=100):
```

Generate points on a line given direction and a point.

direction: (2,) array

point: (2,) array

k1, k2: scalar parameters along the line

num: number of points

Returns: 2 x num array of points

```
k = np.linspace(k1, k2, num)
```

```
line_pts = point.reshape(2,1) + direction.reshape(2,1) * k
```

```
return line_pts
```

```
# Slopes
```

```
m = 1
```

```
n = -1
```

# Only Python code

```
# Normals (for line:  $y = m x + c \Rightarrow -m x + y = c$ )
n1 = np.array([-m, 1])
n2 = np.array([-m, 1])
n3 = np.array([-n, 1])
n4 = np.array([-n, 1])

# Constants c
c1 = 0
c2 = 1
c3 = 0
c4 = 1

# Find intersection points
P1 = line_intersect(n1, c1, n3, c3)
P2 = line_intersect(n1, c1, n4, c4)
P3 = line_intersect(n2, c2, n4, c4)
P4 = line_intersect(n2, c2, n3, c3)

# Parallelogram points array
```

# Only Python code

```
# Calculate area via cross product
vec1 = P2 - P1
vec2 = P4 - P1
area = abs(np.cross(vec1.flatten(), vec2.flatten()))

# Direction vectors for lines (x direction)
dir_m = np.array([1, m])
dir_n = np.array([1, n])

# Generate line points for plotting
k1, k2 = -5, 5

line_m0 = line_dir_pt(dir_m, np.array([0,0]), k1, k2)
line_m1 = line_dir_pt(dir_m, np.array([0,1]), k1, k2)
line_n0 = line_dir_pt(dir_n, np.array([0,0]), k1, k2)
line_n1 = line_dir_pt(dir_n, np.array([0,1]), k1, k2)
```

# Only Python code

```
# Plotting
plt.figure(figsize=(8,8))
plt.plot(line_m0[0], line_m0[1], label='y = m x', color='blue')
plt.plot(line_m1[0], line_m1[1], label='y = m x + 1', linestyle='
    --', color='blue')
plt.plot(line_n0[0], line_n0[1], label='y = n x', color='red')
plt.plot(line_n1[0], line_n1[1], label='y = n x + 1', linestyle='
    --', color='red')

plt.plot(parallelogram[0], parallelogram[1], 'k-', linewidth=2,
    label='Parallelogram')
plt.fill(parallelogram[0], parallelogram[1], 'grey', alpha=0.3)

for i, P in enumerate([P1, P2, P3, P4], 1):
    plt.plot(P[0], P[1], 'ko')
    plt.text(P[0] + 0.1, P[1] + 0.1, f'P{i}', fontsize=12)
```

# Only Python code

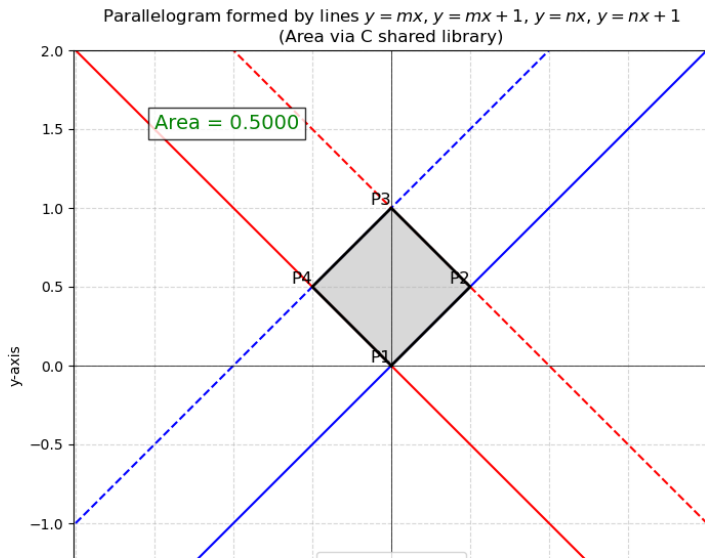
```
plt.text(-4.5, 4, f'Area = {area:.3f}', fontsize=14, color='green',
        ,
        bbox=dict(facecolor='white', alpha=0.8))

plt.axhline(0, color='black', lw=0.5)
plt.axvline(0, color='black', lw=0.5)
plt.grid(True, linestyle='--', alpha=0.6)
plt.axis('equal')
```

# Only Python code

```
plt.xlim(-5, 5)
plt.ylim(-5, 5)
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.title('Parallelogram formed by lines  $y=mx$ ,  $y=mx+1$ ,  $y=nx$ ,  $y=nx+1$ ')
plt.legend()
plt.show()
```

# PLOTS



# PLOTS

