4.13.18

Bhargav - EE25BTECH11013

September 13, 2025

Question

The line L is given by $\frac{x}{5} + \frac{y}{b} = 1$ passes through the point $(13, 32)$. The line K is parallel to L and has the equation $\frac{x}{c} + \frac{y}{3} = 1$. Find the distance between L and K.

Equation of Line L

General form:

$$\mathbf{n}^T\mathbf{x} = 1 \tag{1}$$

For line L:

$$\begin{pmatrix} \frac{1}{5} & \frac{1}{b} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 1 \tag{2}$$

Since it passes through $\begin{pmatrix} 13 \\ 32 \end{pmatrix}$

$$\frac{13}{5} + \frac{32}{b} = 1 \implies b = -20. \tag{3}$$

Equation of Line K

Equation of line K:

$$\begin{pmatrix} \frac{1}{c} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 1 \tag{4}$$

Since $K \parallel L$:

$$\begin{pmatrix} \frac{1}{c} \\ \frac{1}{3} \end{pmatrix} = \lambda \begin{pmatrix} \frac{1}{5} \\ -\frac{1}{20} \end{pmatrix} \tag{5}$$

Thus,

$$\lambda = -\frac{20}{3}, \quad c = -\frac{3}{4}. \tag{6}$$

Normal Vector

The normal vector is:

$$\mathbf{n} = \begin{pmatrix} 4 \\ -1 \end{pmatrix} \tag{7}$$

Distance between parallel lines:

$$\text{Distance} = \frac{|c_1 - c_2|}{\|\mathbf{n}\|} \tag{8}$$

Distance Calculation

Substituting values:

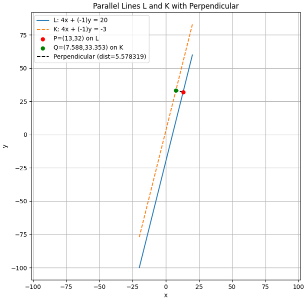
$$\text{Distance} = \frac{|20 - (-3)|}{\sqrt{4^2 + (-1)^2}} \tag{9}$$

$$= \frac{23}{\sqrt{17}}. \tag{10}$$

Therefore, the distance between the two parallel lines is:

$$\frac{23}{\sqrt{17}} \tag{11}$$

Figure


Parallel Lines L and K with Perpendicular
L: 4x + (-1)y = 20
K: 4x + (-1)y = -3
● P=(13,32) on L
● Q=(7.588,33.353) on K
Perpendicular (dist=5.578319)
75
50
25
0
−25
−50
−75
−100
y
−100
−75
−50
−25
0
25
50
75
100
x

C Code

```c
#include <stdio.h>
#include <math.h>

double distance_between_lines(double c1, double c2, double nx,
    double ny) {
    double denom = sqrt(nx * nx + ny * ny);
    return fabs(c1 - c2) / denom;
}

void projection_on_line(double px, double py, double nx, double
    ny, double c,
                        double* qx, double* qy) {
    double denom = nx * nx + ny * ny;
    double t = (c - (nx * px + ny * py)) / denom;
    *qx = px + nx * t;
    *qy = py + ny * t;
}
```

Python + C Code

```
import numpy as np
import matplotlib.pyplot as plt
import ctypes
lib = ctypes.CDLL(./liblines.so)
lib.distance_between_lines.argtypes = [
    ctypes.c_double, ctypes.c_double, ctypes.c_double, ctypes.
        c_double
]
lib.distance_between_lines.restype = ctypes.c_double
lib.projection_on_line.argtypes = [
    ctypes.c_double, ctypes.c_double, ctypes.c_double, ctypes.
        c_double, ctypes.c_double,
    ctypes.POINTER(ctypes.c_double), ctypes.POINTER(ctypes.
        c_double),
]
lib.projection_on_line.restype = None
```

Python + C Code

```
dist = lib.distance_between_lines(c1, c2, nx, ny)
print(fDistance between L and K = {dist:.6f})

qx, qy = ctypes.c_double(), ctypes.c_double()
lib.projection_on_line(px, py, nx, ny, c2, ctypes.byref(qx),
    ctypes.byref(qy))
Q = (qx.value, qy.value)
print(fFoot of perpendicular Q on K = {Q})

x_vals = np.linspace(-20, 20, 400)
y_L = (nx * x_vals - c1) / ny # line L: 4x - y = 20
y_K = (nx * x_vals - c2) / ny # line K: 4x - y = -3

plt.figure(figsize=(8, 8))
plt.plot(x_vals, y_L, label=Line L: 4x - y = 20)
plt.plot(x_vals, y_K, --, label=Line K: 4x - y = -3)
```

Python + C Code

```
plt.scatter(px, py, color=red, zorder=5, label=fP=({int(px)},{int(py)}) on L)
plt.scatter(*Q, color=green, zorder=5, label=fQ=({Q[0]:.2f},{Q[1]:.2f}) on K)
plt.plot([px, Q[0]], [py, Q[1]], k--, linewidth=1.5,
         label=fPerpendicular (dist={dist:.3f}))
plt.xlabel(x)
plt.ylabel(y)
plt.title(Parallel Lines L and K with Perpendicular)
plt.legend()
plt.grid(True)
plt.axis(equal)
plt.savefig(/Users/bhargavkrish/Desktop/BackupMatrix/ee25btech11013/matgeo/4.13.18/figs/Figure_1.png)
plt.show()
```

Python Code

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

def distance_between_lines(c1, c2, nx, ny):
    return abs(c1 - c2) / np.hypot(nx, ny)

def projection_on_line(px, py, nx, ny, c):
    denom = nx*nx + ny*ny
    t = (c - (nx*px + ny*py)) / denom
    qx = px + nx*t
    qy = py + ny*t
    return qx, qy

px, py = 13.0, 32.0
a = 5.0
b = py / (1 - px / a)
```

Python Code

```
print(Computed b:, b)
nL = np.array([1.0/a, 1.0/b])
scale = 20.0
nx, ny = (nL * scale).tolist()
c1 = 1.0 * scale
c2 = -3.0
dist = distance_between_lines(c1, c2, nx, ny)
qx, qy = projection_on_line(px, py, nx, ny, c2)
print(Distance between L and K:, dist)
print(Foot Q on K:, (qx, qy))
x_vals = np.linspace(-20, 20, 400)
# CORRECT formula: y = (c - nx*x) / ny
y_L = (c1 - nx * x_vals) / ny
y_K = (c2 - nx * x_vals) / ny
plt.figure(figsize=(8, 8))
```

Python Code

```
plt.plot(x_vals, y_L, label=fL: {int(nx)}x + ({int(ny)})y = {int(c1)})
plt.plot(x_vals, y_K, --, label=fK: {int(nx)}x + ({int(ny)})y = {int(c2)})
plt.scatter(px, py, color=red, zorder=5, label=fP=({int(px)},{int(py)}) on L)
plt.scatter(qx, qy, color=green, zorder=5, label=fQ=({qx:.3f},{qy:.3f}) on K)
plt.plot([px, qx], [py, qy], k--, linewidth=1.5, label=fPerpendicular (dist={dist:.6f}))
plt.xlabel(x); plt.ylabel(y)
plt.title(Parallel Lines L and K with Perpendicular)
plt.axis(equal)
plt.grid(True); plt.legend()
plt.savefig('/Users/bhargavkrish/Desktop/BackupMatrix/ee25btech11013/matgeo/4.13.18/figs/Figure_1.png')
plt.show()
```