

## 4.2.22

EE25BTECH11019 – Darji Vivek M.

## Question:

Show that the two lines

$$a_1x + b_1y + c_1 = 0, \quad a_2x + b_2y + c_2 = 0$$

where  $b_1b_2 \neq 0$  are parallel if

$$\frac{a_1}{b_1} = \frac{a_2}{b_2}.$$

# Solution

Normal vector of first line:

$$\mathbf{n}_1 = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}$$

Normal vector of second line:

$$\mathbf{n}_2 = \begin{pmatrix} a_2 \\ b_2 \end{pmatrix}$$

If

$$\frac{a_1}{b_1} = \frac{a_2}{b_2},$$

then

$$a_1 b_2 - a_2 b_1 = 0$$

which can be written as

$$a_1 b_2 - a_2 b_1 = \det \begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \end{pmatrix}$$

Thus,

$$\det(\mathbf{n}_1, \mathbf{n}_2) = 0$$

which means  $\mathbf{n}_1$  and  $\mathbf{n}_2$  are collinear, i.e.

$$\mathbf{n}_1 = \lambda \mathbf{n}_2, \quad \lambda \in \mathbb{R}.$$

Since normals are scalar multiples, the direction vectors of the two lines are also scalar multiples.

# Conclusion

Therefore, the two lines have identical slopes and are parallel whenever

$$\frac{a_1}{b_1} = \frac{a_2}{b_2}, \quad b_1 b_2 \neq 0.$$

# Pyhton plot

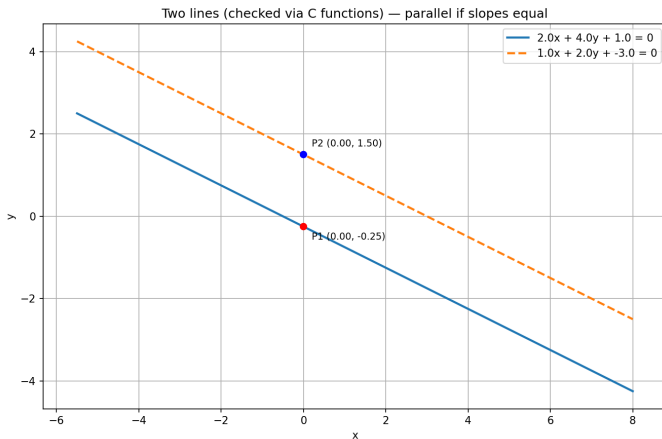


Figure: parallel lines

# C Code: parallel\_funcs.c

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

#define EPS 1e-9

// Return 1 if lines are parallel, else 0
int is_parallel(double a1, double b1, double a2, double b2) {
    double det = a1*b2 - a2*b1;
    if (fabs(det) < EPS) return 1;
    return 0;
}

// Evaluate line  $a*x + b*y + c = 0$ 
void eval_line(double a, double b, double c,
               double *x_in, double *y_out, int n) {
    for (int i = 0; i < n; ++i) {
        y_out[i] = (-a * x_in[i] - c) / b;
    }
}
```



# Python: Load C Library

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

libpath = os.path.join('.', 'libparallel.so')
lib = ctypes.CDLL(libpath)

# Signatures
lib.is_parallel.argtypes = [ctypes.c_double, ctypes.c_double,
                             ctypes.c_double, ctypes.c_double]
lib.is_parallel.restype = ctypes.c_int
lib.eval_line.argtypes = [ctypes.c_double, ctypes.c_double, ctypes.
                           c_double,
                           ctypes.POINTER(ctypes.c_double),
                           ctypes.POINTER(ctypes.c_double),
                           ctypes.c_int]
lib.eval_line.restype = None
```

# Python: Check Parallelism

```
def check_parallel(a1, b1, a2, b2):  
    return bool(lib.is_parallel(a1, b1, a2, b2))  
  
def eval_line(a, b, c, xs):  
    n = len(xs)  
    XTYPE = ctypes.c_double * n  
    x_arr = XTYPE(*xs)  
    y_arr = XTYPE()  
    lib.eval_line(a, b, c, x_arr, y_arr, n)  
    return np.array([y_arr[i] for i in range(n)])  
  
# Example lines  
a1, b1, c1 = 2.0, 4.0, 1.0  
a2, b2, c2 = 1.0, 2.0, -3.0  
print("Are parallel? ->", check_parallel(a1, b1, a2, b2))
```

# Python: Plotting Lines

```
xs = np.linspace(-10, 10, 600)
ys1 = eval_line(a1, b1, c1, xs)
ys2 = eval_line(a2, b2, c2, xs)

plt.figure()
plt.plot(xs, ys1, label=f'{a1}x+{b1}y+{c1}=0')
plt.plot(xs, ys2, '--', label=f'{a2}x+{b2}y+{c2}=0')
plt.legend(); plt.grid(True)
plt.axis('equal')
plt.savefig('parallel_lines_ctypes.png', dpi=150)
plt.show()
```