4.2.22

EE25BTECH11019 – Darji Vivek M.

Question

Question:
Show that the two lines

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

where $b_1 b_2 \neq 0$ are parallel if

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

Solution: Step 1

Form the 2 × 2 coefficient matrix of normals:

$$\mathbf{M} = \begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \end{pmatrix}.$$

Solution: Step 2

Assume

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

Then there exists $k \in \mathbb{R}$ such that

$$a_2 = ka_1, \qquad b_2 = kb_1.$$

Solution: Step 3

Write the rows of M as row vectors:

$$\text{Row}_1 = (a_1,\ a_2) = a_1(1,\ k),$$

$$\text{Row}_2 = (b_1,\ b_2) = b_1(1,\ k).$$

Solution: Step 4

Perform the row operation:

$$\text{Row}_2 \;\rightarrow\; \text{Row}_2 - \frac{b_1}{a_1}\,\text{Row}_1$$

(assuming $a_1 \neq 0$; if $a_1 = 0$, swap roles).
Since Row_2 is a multiple of Row_1, we get

$$\text{Row}_2 = (0, 0).$$

Solution: Step 5

Thus the row-echelon form of M has exactly one nonzero row:

$$\text{rank}(\mathbf{M}) = 1.$$

Rank 1 means the two normal vectors are linearly dependent (collinear). Therefore, the lines have the same slope and are parallel.

Pyhton plot

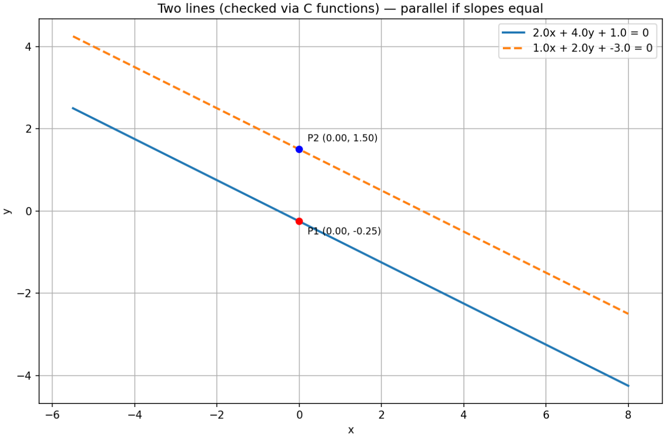

Figure: parallel lines

C Code: parallel_funcs.c

```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

```python
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()
```