2.10.65

M Chanakya Srinivas- EE25BTECH11036

Problem Statement

Let $OACB$ be a parallelogram with O at the origin and OC a diagonal. Let D be the midpoint of OA.

- Prove that BD and CO intersect in the same ratio.
- Determine this ratio.

Step 1: Define Position Vectors

$$\mathbf{O} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \tag{1}$$

Midpoint D of OA:

$$\mathbf{D} = \frac{\mathbf{O} + \mathbf{A}}{2} = \frac{1}{2} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \tag{2}$$

Step 2: Lines in Vector Form

Line BD:

$$\begin{aligned}\mathbf{R_1} &= \mathbf{B} + \lambda(\mathbf{D} - \mathbf{B}) \\ &= \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \end{pmatrix} - \lambda \left(\frac{1}{2} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right) \end{aligned} \tag{3}$$

Line CO:

$$\begin{aligned}\mathbf{R_2} &= \mathbf{C} + \mu(\mathbf{O} - \mathbf{C}) \\ &= (1 - \mu) \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \end{aligned} \tag{4}$$

Step 3: Find Intersection

Equate lines:

$$\begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \end{pmatrix} - \lambda \left(\frac{1}{2} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right) = (1 - \mu) \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \tag{5}$$

Solve coefficients:
For **a**:

$$1 - \frac{\lambda}{2} = 0 \implies \lambda = 2 \tag{6}$$

For **b**:

$$1 - \lambda = 1 - \mu \implies \mu = 2 \tag{7}$$

Step 4: Intersection Ratio

- On BD, $\lambda = 2 \Rightarrow$ intersection divides BD in ratio $2:1$ - On CO, $\mu = 2$ $\Rightarrow$ intersection divides CO in ratio $2:1$

The lines BD and CO intersect in the ratio $2:1$.

C code

```c
// parallelogram.c
#include <stdio.h>
typedef struct {
    double x;
    double y;
} Point;
static Point result;
Point intersection() {
    Point O = {0.0, 0.0};
    Point A = {1.0, 0.0};
    Point B = {0.0, 1.0};
    Point D = {(O.x + A.x)/2.0, (O.y + A.y)/2.0};
    double lam = 2.0/3.0;
    result.x = B.x + lam * (D.x - B.x);
    result.y = B.y + lam * (D.y - B.y);
  return result;
}
```

Python code through shared output

```
import ctypes
import matplotlib.pyplot as plt
import numpy as np
# Load the shared library
lib = ctypes.CDLL(./libparallelogram.so)
# Define return type for the intersection function
class Point(ctypes.Structure):
    _fields_ = [(x, ctypes.c_double), (y, ctypes.c_double)]
lib.intersection.restype = Point
# Get intersection P
P = lib.intersection()
print(Intersection P =, (P.x, P.y))
# Define points
O = np.array([0, 0])
A = np.array([1, 0])
B = np.array([0, 1])
C = A + B
D = (O + A)/2
P_vec = np.array([P.x, P.y])
```

Python code through shared output

```
# --- Plot ---
fig, ax = plt.subplots()
# Parallelogram
ax.plot([O[0], A[0], C[0], B[0], O[0]],
        [O[1], A[1], C[1], B[1], O[1]], 'k-')
# Diagonal OC
ax.plot([O[0], C[0]], [O[1], C[1]], 'r--', label=OC)
# Line BD
ax.plot([B[0], D[0]], [B[1], D[1]], 'g--', label=BD)
# Points
for Pnt, name in zip([O,A,B,C,D,P_vec], ['O','A','B','C','D','P'
    ]):
    ax.scatter(Pnt[0], Pnt[1], s=50)
    ax.text(Pnt[0]+0.05, Pnt[1]+0.05, name)
ax.set_aspect('equal')
ax.legend()
plt.title(Intersection of BD and CO in ratio 2:1)
plt.show()
```

only Python code

```
import sys
sys.path.insert(0, '/sdcard/github/matgeo/codes/CoordGeo') # path
     to CoordGeo
import numpy as np
import matplotlib.pyplot as plt
import subprocess
import shlex
import os # << add this
# Local imports
from line.funcs import line_gen
# Define points as column vectors
O = np.array([[0], [0]])
A = np.array([[1], [0]])
B = np.array([[0], [1]])
C = A + B
D = (O + A) / 2
# Compute intersection P on BD such that P = B + (2/3)(D - B)
lam = 2/3
P = B + lam * (D - B)
```

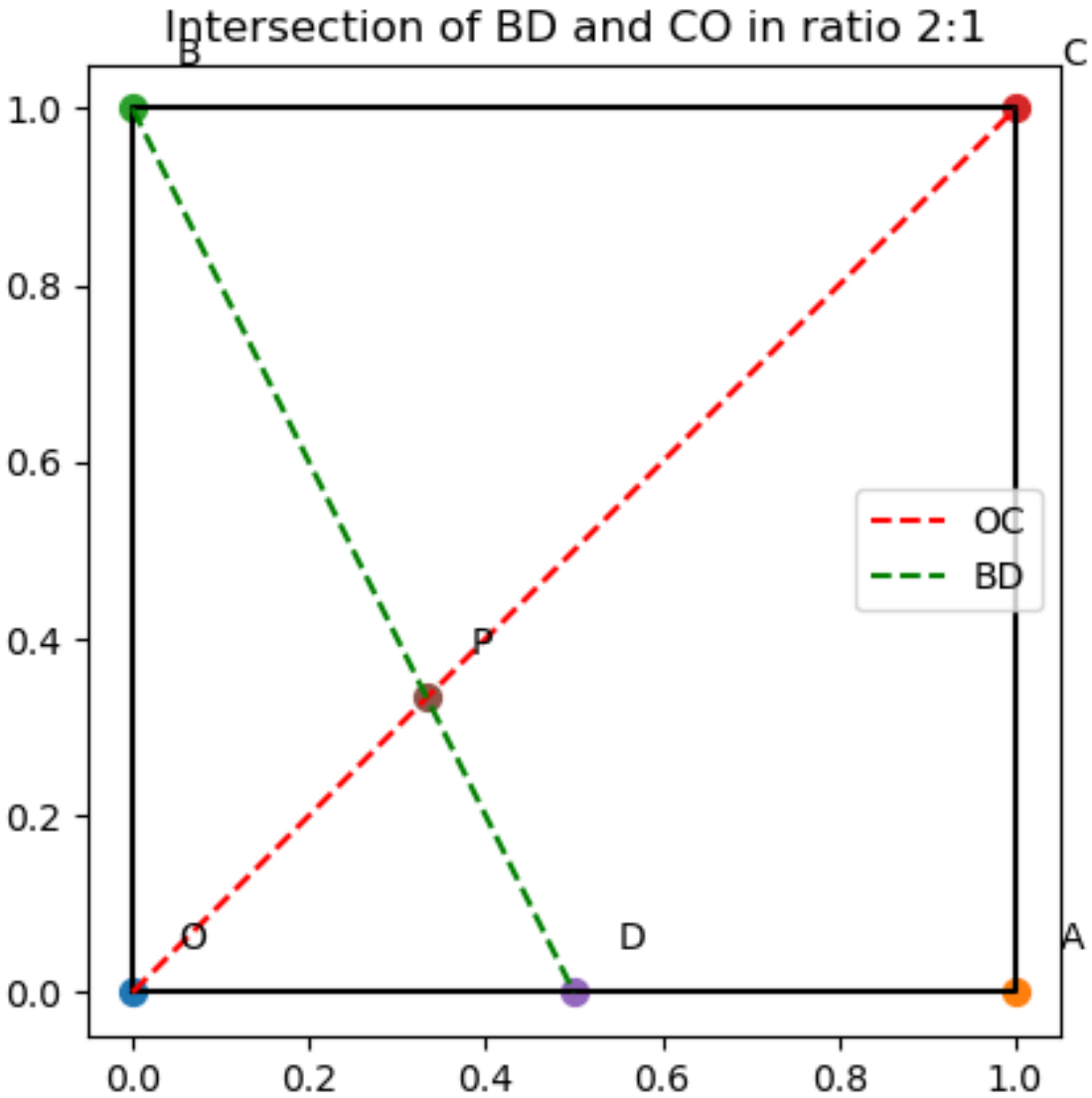
only Python code

```
# Parallelogram edges
x_OA = line_gen(O, A)
x_AB = line_gen(A, C)
x_CB = line_gen(C, B)
x_BO = line_gen(B, O)
plt.plot(x_OC[0,:], x_OC[1,:], 'r--', label='$OC$')
plt.plot(x_BD[0,:], x_BD[1,:], 'g--', label='$BD$')
for line in [x_OA, x_AB, x_CB, x_BO]:
    plt.plot(line[0,:], line[1,:], 'k-')
coords = np.block([[O, A, B, C, D, P]])
labels = ['O', 'A', 'B', 'C', 'D', 'P']
plt.scatter(coords[0,:], coords[1,:])
for i, txt in enumerate(labels):
    plt.annotate(f'{txt}\n({coords[0,i]:.2f}, {coords[1,i]:.2f})'
        ,
                 (coords[0,i], coords[1,i]),
                 textcoords=offset points,
                 xytext=(10,-10),
                 ha='center')
```

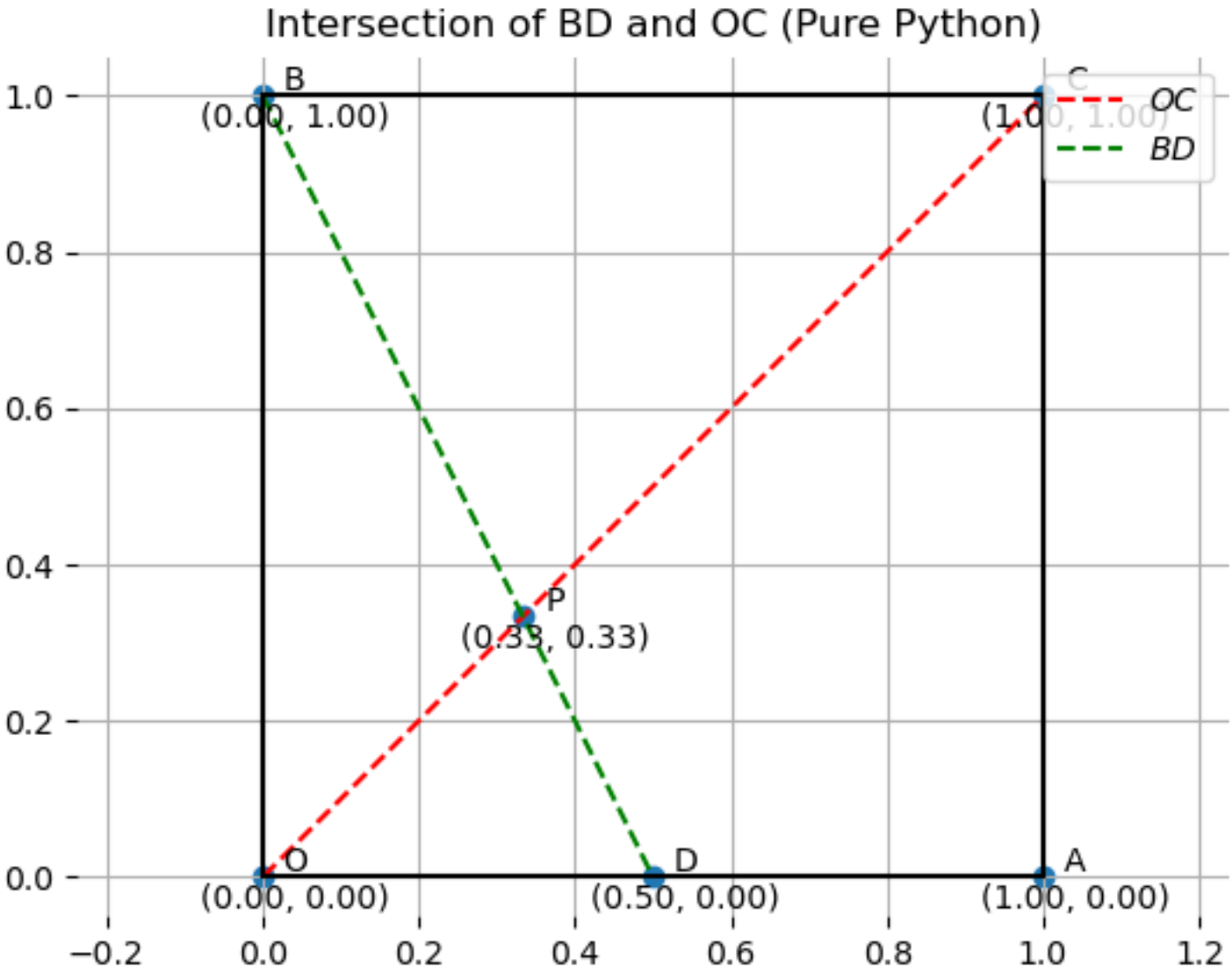
only Python code

```
# Styling
ax = plt.gca()
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['bottom'].set_visible(False)
plt.axis('equal')
plt.grid()
plt.legend(loc='best')
plt.title(Intersection of BD and OC (Pure Python))
# Create directory if needed and save figure
save_path = 'chapters/10/7/2/2/figs/fig.pdf'
os.makedirs(os.path.dirname(save_path), exist_ok=True)
plt.savefig(save_path)
try:
    subprocess.run(shlex.split(ftermux-open {save_path}))
except:
    plt.show()
```

PLOTS


Intersection of BD and CO in ratio 2:1
B
C
A
O
D
P
OC
BD
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

PLOTS


Intersection of BD and OC (Pure Python)
B
(0.00, 1.00)
C
(1.00, 1.00)
OC
BD
P
(0.33, 0.33)
O
(0.00, 0.00)
D
(0.50, 0.00)
A
(1.00, 0.00)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2