1.5.24

M Chanakya Srinivas- EE25BTECH11036

Problem statement

A line intersects the Y-axis and X-axis at the points

$$P = (0, b) \quad \text{and} \quad Q = (c, 0)$$

respectively. If $(2, -5)$ is the midpoint of $\overline{PQ}$, find the coordinates of P and Q.

Idea (two relations needed)

We need two independent relations between b and c. We'll obtain:

1. a relation coming from the *line equation / rank condition*, and
2. a relation coming from the *midpoint* condition.

Both together determine b and c.

Line equation and the rank relation

Consider the general line equation

$$ux + vy + w = 0,$$

with constants u, v, w (not all zero). Since $P = (0, b)$ lies on the line,

$$u \cdot 0 + v \cdot b + w = 0 \quad \Rightarrow \quad vb + w = 0. \tag{1}$$

Since $Q = (c, 0)$ lies on the line,

$$u \cdot c + v \cdot 0 + w = 0 \quad \Rightarrow \quad uc + w = 0. \tag{2}$$

Subtract (2) from (1):

$$vb - uc = 0 \quad \Rightarrow \quad vb = uc. \tag{R}$$

Equation (R) is the relation between b and c arising from the line coefficients. (It is independent of w.)

Interpretation of the rank relation

The relation $vb = uc$ simply says the ratio of the intercepts is tied to the ratio of line coefficients:

$$\frac{b}{c} = \frac{u}{v}.$$

This is one constraint linking b and c. To get their numerical values we need another independent relation — the midpoint condition.

Midpoint condition (second relation)

The midpoint M of P and Q is

$$M = \frac{P+Q}{2} = \frac{1}{2}\begin{pmatrix}0\\b\end{pmatrix} + \frac{1}{2}\begin{pmatrix}c\\0\end{pmatrix} = \frac{1}{2}\begin{pmatrix}c\\b\end{pmatrix}.$$

We are given $M = \begin{pmatrix}2\\-5\end{pmatrix}$. Equate the components:

$$\frac{c}{2} = 2 \quad \Rightarrow \quad c = 4,$$

$$\frac{b}{2} = -5 \quad \Rightarrow \quad b = -10.$$

These are the numerical values of b and c.

Consistency with the rank relation

We found:

$$b = -10, \qquad c = 4.$$

Plugging into the rank relation $vb = uc$ gives

$$v(-10) = u(4) \quad \Rightarrow \quad \frac{u}{v} = -\frac{10}{4} = -\frac{5}{2}.$$

So any line coefficients u, v satisfying $u : v = -5 : 2$ (and appropriate w) will give the same intercepts. This confirms the intercepts are consistent with a line having slope $-b/c = -(-10)/4 = 10/4 = 5/2$ (note sign conventions).

Matrix viewpoint (brief)

Write the midpoint equations componentwise:

$$\frac{1}{2}\begin{pmatrix} c \\ b \end{pmatrix} = \begin{pmatrix} 2 \\ -5 \end{pmatrix} \quad \Longrightarrow \quad \begin{pmatrix} c \\ b \end{pmatrix} = \begin{pmatrix} 4 \\ -10 \end{pmatrix}.$$

Equivalently as a matrix equation $I\mathbf{x} = \mathbf{B}$ with $\mathbf{x} = \begin{pmatrix} b \\ c \end{pmatrix}$ (or ordered as you prefer), the solution is immediate since $I^{-1} = I$.

Final answer

Therefore the intercept points are

$$\boxed{P = (0, -10), \qquad Q = (4, 0).}$$

(These satisfy the midpoint condition and the linear relation from the line coefficients.)

Illustration

figs/fig.png

C Code - Section formula function

```c
#include <stdio.h>

// Function to find coordinates of P(0,b) and Q(c,0)
// given the midpoint (mx, my)
void findCoordinates(int mx, int my, int* c, int* b) {

    printf(Step 1: Rank relation between b and c\n);
    printf(Line: ux + vy + w = 0\n);
    printf(P = (0,b), Q = (c,0) lie on the line\n);
    printf(=> vb + w = 0, uc + w = 0\n);
    printf(Subtracting: vb - uc = 0 => v b = u c (relation 1)\n\n
        );

    printf(Step 2: Midpoint relation\n);
    printf(Midpoint M = ( (0+c)/2, (b+0)/2 ) = (%d, %d)\n, mx, my
        );
    printf(=> c/2 = %d => c = %d\n, mx, 2*mx);
    printf(=> b/2 = %d => b = %d\n\n, my, 2*my);
```

C Code - Section formula function

```
    // Assign values using midpoint
    *c = 2 * mx;
    *b = 2 * my;

    printf(Step 3: Solve both relations\n);
    printf(Coordinates of P = (0,%d)\n, *b);
    printf(Coordinates of Q = (%d,0)\n\n, *c);
}
```

Python Code through shared output

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

# --- Step 1: Rank relation ---
print(Step 1: Rank relation between b and c)
print(General line: ux + vy + w = 0)
print(Points: P=(0,b), Q=(c,0))
print(Substitute: vb + w = 0, uc + w = 0)
print(Subtracting => v b = u c (relation 1)\n)

# --- Step 2: Midpoint relation ---
print(Step 2: Midpoint relation)
mx, my = 2, -5
print(fMidpoint M = ((0+c)/2, (b+0)/2) = ({mx}, {my}))

# Solve midpoint equations
c = 2 * mx
b = 2 * my
print(fc/2 = {mx} => c = {c})
```

Python Code through shared output

```
print(fb/2 = {my} => b = {b} (relation 2)\n)

# --- Step 3: Solve ---
P = (0, b)
Q = (c, 0)
M = (mx, my)

print(Step 3: Solve both relations)
print(fCoordinates of P = {P})
print(fCoordinates of Q = {Q}\n)

# --- Step 4: Plot ---
x_points = np.array([P[0], Q[0]])
y_points = np.array([P[1], Q[1]])

plt.figure(figsize=(7,6))
plt.plot(x_points, y_points, 'b-', label=Line PQ)
plt.plot(P[0], P[1], 'go', markersize=10, label=fP {P})
plt.plot(Q[0], Q[1], 'ro', markersize=10, label=fQ {Q})
```

Python Code through shared output

```
plt.plot(M[0], M[1], 'm*', markersize=12, label=fM {M})

# Annotate
plt.text(P[0]+0.2, P[1], fP{P}, fontsize=12)
plt.text(Q[0]+0.2, Q[1], fQ{Q}, fontsize=12)
plt.text(M[0]+0.2, M[1], fM{M}, fontsize=12)

# Format
plt.title(Line Intercepting Axes with Midpoint Condition,
    fontsize=15)
plt.axhline(0, color=black, linewidth=0.7)
plt.axvline(0, color=black, linewidth=0.7)
plt.xlabel(X-axis)
plt.ylabel(Y-axis)
plt.grid(True, linestyle=--, alpha=0.6)
plt.legend()
plt.axis(equal)
plt.show()
```

Direct Python Code

```
import sys # for path to external scripts
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt

# local imports
from libs.line.funcs import *
from libs.triangle.funcs import *
from libs.conics.funcs import circ_gen

# --- Step 1: Rank relation ---
print(Step 1: Rank relation between b and c)
print(General line: ux + vy + w = 0)
print(Points: P = (0,b), Q = (c,0))
print(=> vb + w = 0, uc + w = 0)
print(Subtracting => v b = u c (Relation 1)\n)

# --- Step 2: Midpoint relation ---
print(Step 2: Midpoint relation)
```

Direct Python Code

```
M = np.array(([2, -5])).reshape(-1,1)
print(Midpoint M = ((0+c)/2, (b+0)/2) = (2, -5))
c = 2 * M[0,0]
b = 2 * M[1,0]
print(fc/2 = 2 => c = {c})
print(fb/2 = -5 => b = {b} (Relation 2)\n)

# --- Step 3: Solve both relations ---
P = np.array(([0,b])).reshape(-1,1)
Q = np.array(([c,0])).reshape(-1,1)

print(Step 3: Solve both relations)
print(fCoordinates of P = (0, {b}))
print(fCoordinates of Q = ({c}, 0)\n)

# --- Step 4: Plotting ---
x_PQ = line_gen(P,Q)
plt.plot(x_PQ[0,:], x_PQ[1,:], label='$PQ$')
```

Direct Python Code

```
# Mark points
coords = np.block([[P,Q,M]])
vert_labels = ['P','Q','M']
plt.scatter(coords[0,:], coords[1,:], color=['green','red','
    magenta'])
for i, txt in enumerate(vert_labels):
    plt.annotate(f'{txt}\n({coords[0,i]:.0f},{coords[1,i]:.0f})',
                 (coords[0,i], coords[1,i]),
                 textcoords=offset points, xytext=(20,-10), ha='
                     center')

# Axis 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.legend(loc='best')
```

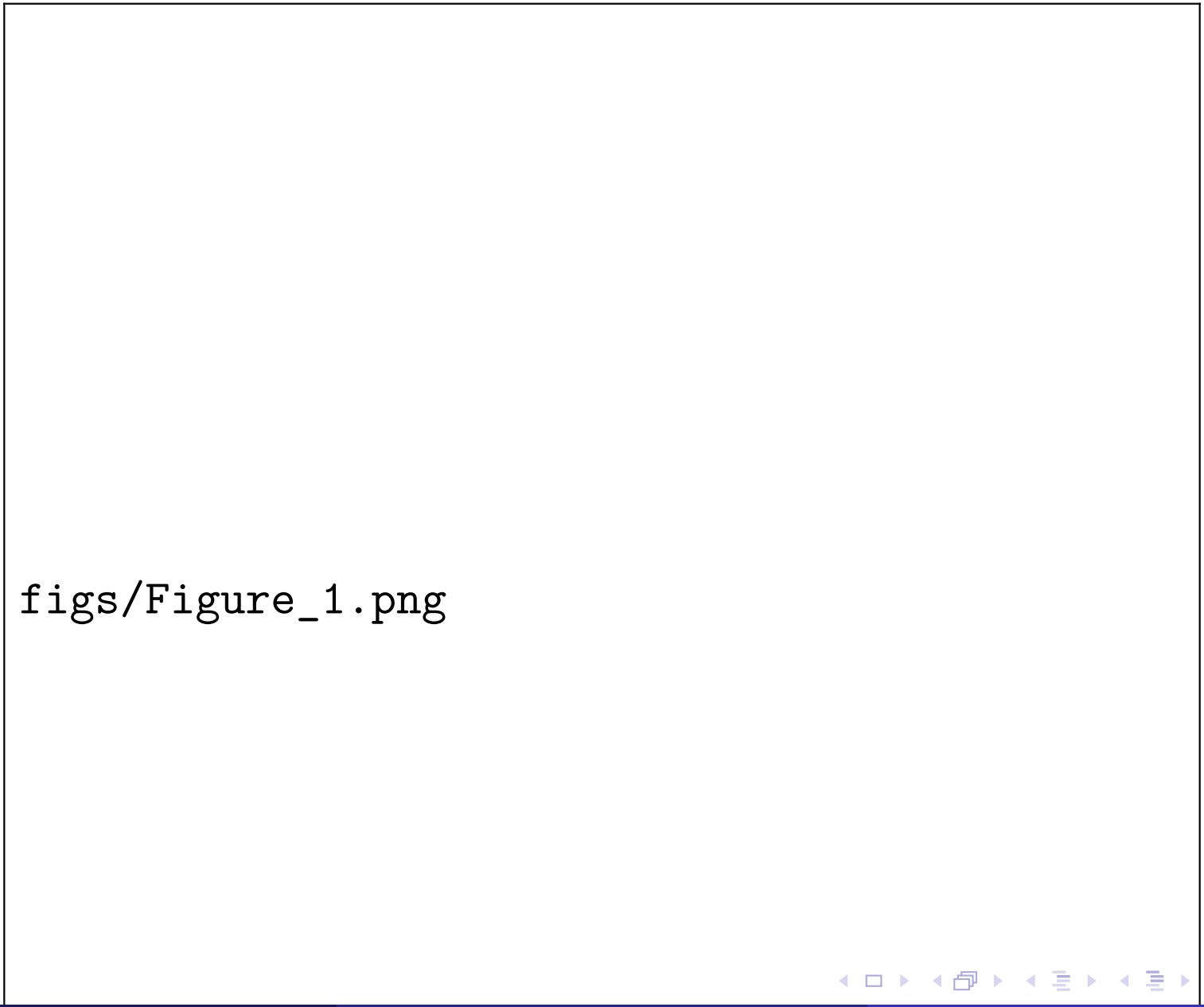
Direct Python Code

```
# Save figure as PDF
outfile_pdf = 'chapters/10/7/2/2/figs/fig.pdf'
plt.savefig(outfile_pdf)

# Save figure as PNG
outfile_png = 'chapters/10/7/2/2/figs/fig.png'
plt.savefig(outfile_png, dpi=300)

# Open image depending on system
try:
    import platform, subprocess, shlex
    if termux in platform.platform().lower(): # Android Termux
        subprocess.run(shlex.split(ftermux-open {outfile_png}))
    else: # Linux desktop
        subprocess.run(shlex.split(fxdg-open {outfile_png}))
except Exception as e:
    print(fCould not auto-open file. Saved at {outfile_png})
```

Plot by python using shared output from c



Plot by python only

figs/fig.png