1.5.29

Nipun Dasari - EE25BTECH11042

September 12, 2025

Question

The coordinates of the point P dividing the line segment joining the points A (1, 3) and B (4, 6), in the ratio 2 : 1 are

Equation

The formula for internal division of vectors is where P divides A and B in the ratio k:1

$$\mathsf{P} = \mathsf{kB{+}A}\frac{}{1+k}$$

Theoretical Solution

According to the question,
Consider the coordinate as following vectors

Given the points A and B

$$\mathbf{A} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 4 \\ 6 \end{pmatrix}$$

Thus by formula

$$\mathbf{P} = \frac{1}{1+2}\left(\begin{pmatrix} 1 \\ 3 \end{pmatrix} + 2\begin{pmatrix} 4 \\ 6 \end{pmatrix}\right)$$

$$\mathbf{P} = \frac{1}{3}\left(\begin{pmatrix} 1 \\ 3 \end{pmatrix} + \begin{pmatrix} 8 \\ 12 \end{pmatrix}\right)$$

$$\mathbf{P} = \frac{1}{3}\begin{pmatrix} 9 \\ 15 \end{pmatrix}$$

C Code - Section formula function

```c
// section_formula.c
#include <stdio.h>

void find_section_point(double x1, double y1, double x2, double
    y2, double m, double n, double* x, double* y) {
        *x = (m * x2 + n * x1) / (m + n);
        *y = (m * y2 + n * y1) / (m + n);
}
```

Python Code through shared output

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

# Load the shared library
lib = ctypes.CDLL(./line_division.so)

# Define argument and return types
lib.divide_point.argtypes = [ctypes.c_float, ctypes.c_float,
    ctypes.c_float, ctypes.c_float,
ctypes.c_float, ctypes.c_float,
np.ctypeslib.ndpointer(dtype=np.float32, ndim=1, flags=
    C_CONTIGUOUS)]
lib.divide_point.restype = None

# Given points A and B
x1, y1 = 1.0, 3.0 # Point A
x2, y2 = 4.0, 6.0 # Point B
```

Python Code through shared output

```python


# Hardcoded ratio m:n
m, n = 2.0, 1.0

# Output array to store coordinates of P
out = np.zeros(2, dtype=np.float32)

# Call the C function to get point P
lib.divide_point(x1, y1, x2, y2, m, n, out)
Px, Py = out[0], out[1]

# Print the coordinates of P
print(fCoordinates of P dividing AB in {m}:{n} ratio: ({Px}, {Py
    }))
```

Python Code through shared output

```python

# Plot A, B, and P
plt.figure()
plt.plot([x1, x2], [y1, y2], 'k--', label=Line AB) # line AB
plt.scatter([x1, x2, Px], [y1, y2, Py], color=['red','blue','green'], label=Points)
plt.text(x1, y1,  A, fontsize=10)
plt.text(x2, y2,  B, fontsize=10)
plt.text(Px, Py,  P, fontsize=10)
plt.legend([Line AB,Points])
plt.xlabel(X)
plt.ylabel(Y)
plt.title(fPoint dividing AB in {m}:{n} ratio)
plt.grid(True)
plt.show()
```

Python code : Direct

```
import sys
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

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



#Given points
A = np.array(([1,3])).reshape(-1,1)
B = np.array(([4,6])).reshape(-1,1)

#Ratio
n=2/1
```

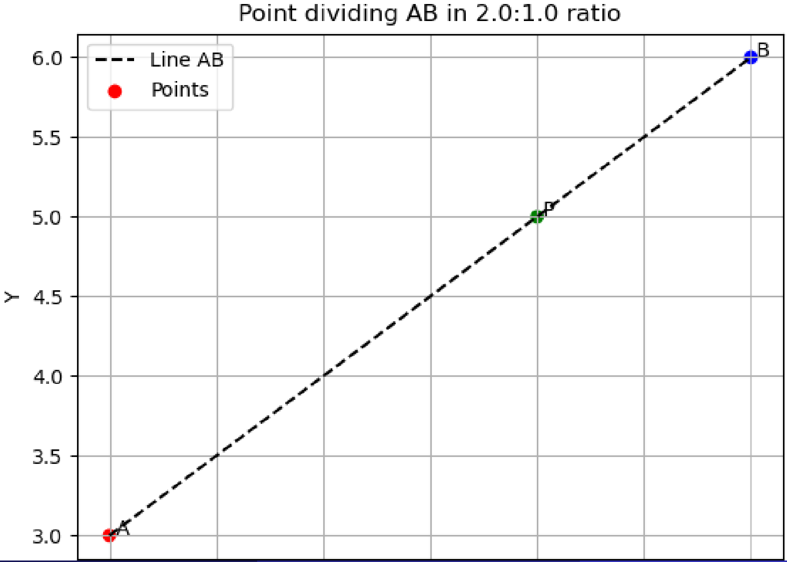
Python code : Direct

```python
#Point
P= (A+n*B)/(1+n) # calculating the coordinate points of R which divides the join between the two points
#print(R)

#Generating all lines
x_AB = line_gen(A,B)

#Plotting all lines
plt.plot(x_AB[0,:],x_AB[1,:],label='$AB$')

#Labeling the coordinates
tri_coords = np.block([[A,B,P]])
plt.scatter(tri_coords[0,:], tri_coords[1,:])
vert_labels = ['A','B','P']
for i, txt in enumerate(vert_labels):
#plt.annotate(txt, # this is the text
plt.annotate(f'{txt}\n({tri_coords[0,i]:.0f}, {tri_coords[1,i]:.0f})',
```

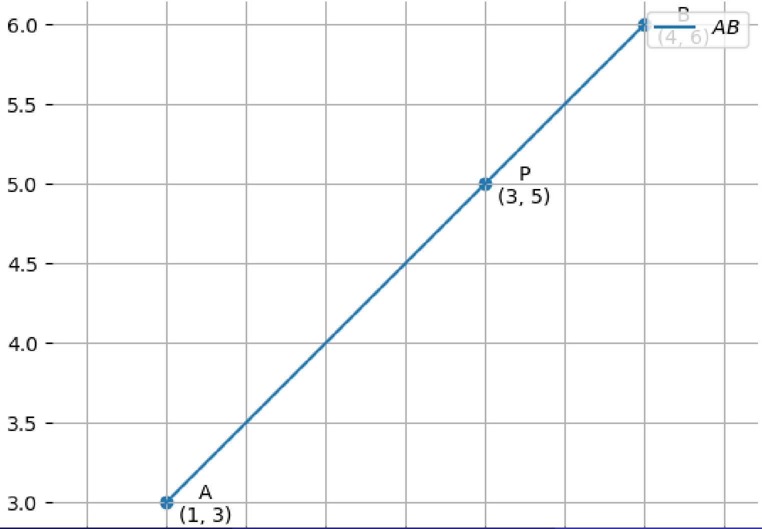
Python code : Direct

```
# use set_position
ax = plt.gca()
#ax.spines['top'].set_color('none')
#ax.spines['left'].set_position('zero')
#ax.spines['right'].set_color('none')
#ax.spines['bottom'].set_position('zero')
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.xlabel('$x$')
#plt.ylabel('$y$')
plt.legend(loc='best')
plt.grid() # minor
plt.axis('equal')
plt.show()
```

Plot by python using shared output from c



Plot by python only


B
(4, 6)
AB
P
(3, 5)
A
(1, 3)