4.13.37

Kartik Lahoti - EE25BTECH11032

September 17, 2025

Question

If x_1, x_2, x_3 as well as y_1, y_2, y_3, are in G.P with the same common ratio then then points (x_1, y_1), (x_2, y_2) and (x_3, y_3)

1. lie on a straight line
2. lie on ellipse
3. lie on circle
4. are vertices of a triangle

Theoretical Solution

Symbol	Value	Description
A	$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$	Given Point
B	$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$	Given Point
C	$\begin{pmatrix} x_3 \\ y_3 \end{pmatrix}$	Given Point

Table: 4.13.37

Theoretical Solution

To check if **A** , **B** and **C** lie on a straight line ,

$$rank\begin{pmatrix}\mathbf{B}-\mathbf{A} & \mathbf{C}-\mathbf{B}\end{pmatrix} = 1 \tag{1}$$

If r is the common ratio for the G.P , then vector **B** and **C** can also be written as

$$\mathbf{B} = r\mathbf{A} \qquad \mathbf{C} = r^2\mathbf{A} \tag{2}$$

Theoretical Solution

$$rank\begin{pmatrix} r\mathbf{A}-\mathbf{A} & r^2\mathbf{A}-r\mathbf{A} \end{pmatrix} = 1 \tag{3}$$

Case 1 : $x_1 \neq 0$

$$(r-1)\begin{pmatrix} x_1 & rx_1 \\ y_1 & ry_1 \end{pmatrix} \xleftrightarrow{R_2 \to R_2 - \frac{y_1}{x_1}R_1} \begin{pmatrix} x_1 & rx_1 \\ 0 & 0 \end{pmatrix} \tag{4}$$

Theoretical Solution

Case 2: $(x_1 = 0 \text{ and } y_1 \neq 0)$ or $(x_1 \neq 0 \text{ and } y_1 = 0)$

$$\begin{pmatrix} 0 & 0 \\ y_1 & ry_1 \end{pmatrix} \quad or \quad \begin{pmatrix} x_1 & rx_1 \\ 0 & 0 \end{pmatrix} \tag{5}$$

From Case 1 and Case 2 we can see $rank = 1$. Thus, the points lie on a straight line

Hence, Answer : (1)

C Code (1)

```
void gp( double *X , double *Y , double fact )
{
        for(int i= 0 ; i <2 ; i++)
                        Y[i] = X[i] * fact;
}
```

C Code (2) - Function to Generate Points on Line

```c
void linegen(double *XY, double *A , double *B , int n , int m )
{
    double temp[m] ;
    for (int i = 0 ; i < m ; i++)
    {
        temp [ i ] = (B[i]- A[i]) /(double) n ;
    }
    for (int i = 0 ; i < n ; i++ )
        for (int j = 0 ; j < m ; j++)
            XY[j*n + i ] = A[j] + temp[j] * i ;

}
```

Python Code - Using Shared Object

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

handc1 = ct.CDLL("./func.so")

handc1.gp.argtypes = [
    ct.POINTER(ct.c_double),
    ct.POINTER(ct.c_double),
    ct.c_double
]

handc1.gp.restype = None
```

Python Code - Using Shared Object

```
O = np.zeros(2 , dtype = np.float64).reshape(-1,1)
A = np.array([1,2] , dtype = np.float64).reshape(-1,1)
B = np.zeros(2 , dtype = np.float64).reshape(-1,1)
C = np.zeros(2, dtype = np.float64).reshape(-1,1)
r = 3
handc1.gp(
    A.ctypes.data_as(ct.POINTER(ct.c_double)),
    B.ctypes.data_as(ct.POINTER(ct.c_double)),
    r)
handc1.gp(
    A.ctypes.data_as(ct.POINTER(ct.c_double)),
    C.ctypes.data_as(ct.POINTER(ct.c_double)),
    r**2)
print("Vector A = ",A)
print("Vector B = " , B)
print("Vector C = " , C)
```

Python Code - Using Shared Object

```
def line(P: np.ndarray , Q: np.ndarray, str1 , str2):
    handc2 = ct.CDLL("./line_gen.so")

    handc2.linegen.argtypes = [
        ct.POINTER(ct.c_double),
        ct.POINTER(ct.c_double),
        ct.POINTER(ct.c_double),
        ct.c_int , ct.c_int
    ]
    handc2.linegen.restype = None
```

Python Code - Using Shared Object

```
n = 200
XY = np.zeros((2,n),dtype=np.float64)

handc2.linegen (
    XY.ctypes.data_as(ct.POINTER(ct.c_double)),
    P.ctypes.data_as(ct.POINTER(ct.c_double)),
    Q.ctypes.data_as(ct.POINTER(ct.c_double)),
    n,2
)
plt.plot(XY[0,:],XY[1,:], str1 , label = str2 )
```

Python Code - Using Shared Object

```
plt.figure()

line(O,A,"g-"," Line Segment : OA ")
line(A,B,"r-"," Line Segment : AB ")
line(B,C,"b-"," Line Segment : BC ")

coords = np.block([[A,B,C,O]])

plt.scatter(coords[0,:] , coords[1,:])
vert_label = ['A','B','C','O']
```

Python Code - Using Shared Object

```
for i , txt in enumerate(vert_label) :

    if i != 2 :
        plt.annotate(f"{txt}\n({coords[0,i]:.1f},{coords[1,i]:.1f
            })",
                    (coords[0,i], coords[1,i]),
                    textcoords = "offset points" ,
                    xytext = (0,12),ha = "center")
    else :
        plt.annotate(f"{txt}\n({coords[0,i]:.1f},{coords[1,i]:.1f
            })",
                    (coords[0,i], coords[1,i]),
                    textcoords = "offset points" ,
                    xytext = (0,-25),ha = "center")
```

Python Code - Using Shared Object

```
plt.xlabel("$x$")
plt.ylabel("$y$")
plt.grid()

plt.legend(loc="best")

plt.title("4.13.37")

plt.savefig("../figs/colli1.png")
plt.show()

#plt.savefig('../figs/colli1.png')
#subprocess.run(shlex.split("termux-open ../figs/colli1.png"))
```

Python Code

```
mport math
import sys
sys.path.insert(0, '/home/kartik-lahoti/matgeo/codes/CoordGeo')
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt

from line.funcs import *

#if using termux
#import subprocess
#import shlex
```

Python Code

```
O = np.zeros(2).reshape(-1,1)
A = np.array([1,2]).reshape(-1,1)
B = np.zeros(2).reshape(-1,1)
C = np.zeros(2).reshape(-1,1)
r = 3
B = r * A
C = (r ** 2 ) *A

print("Vector A = " , A)
print("Vector B = " , B)
print("Vector C = " , C)
```

Python Code

```
def plot_it(P,Q,str1,str2):
    x_l = line_gen_num(P,Q,20)
    plt.plot(x_l[0,:],x_l[1,:] , str1 , label = str2)

plt.figure()
plot_it(O,A,"g-"," Line Segment : OA ")
plot_it(A,B,"r-"," Line Segment : AB ")
plot_it(B,C,"b-"," Line Segment : BC ")

coords = np.block([[A,B,C,O]])

plt.scatter(coords[0,:] , coords[1,:])
vert_label = ['A','B','C','O']
```

Python Code

```
for i , txt in enumerate(vert_label) :

    if i != 2 :
        plt.annotate(f"{txt}\n({coords[0,i]:.1f},{coords[1,i]:.1f
            })",
                    (coords[0,i], coords[1,i]),
                    textcoords = "offset points" ,
                    xytext = (0,12),ha = "center")
    else :
        plt.annotate(f"{txt}\n({coords[0,i]:.1f},{coords[1,i]:.1f
            })",
                    (coords[0,i], coords[1,i]),
                    textcoords = "offset points" ,
                    xytext = (0,-25),ha = "center")
```

Python Code

```
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.grid()
plt.legend(loc = "best")
plt.title("Fig:4.13.37")
plt.savefig("../figs/colli2.png")
plt.show()

#plt.savefig('../figs/colli2.png')
#subprocess.run(shlex.split("termux-open ../figs/colli2.png"))
```

Plot

Taking an example as $\mathbf{A} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and $r = 3$, we get the following graph.