4.3.12

Vishwambhar - EE25BTECH11025

12th september, 2025

Question

Check which of the following are solutions of the equation $x - 2y = 4$ and which are not

1. $(0, 2)$
2. $(2, 0)$
3. $(4, 0)$
4. $\left(\sqrt{2}, 4\sqrt{2}\right)$
5. $(1, 1)$

Given

Given line equation can be written as:

$$\mathbf{n}^\top \mathbf{x} = c \tag{1}$$

where $\mathbf{n} = \begin{pmatrix} 1 \\ -2 \end{pmatrix}$, $\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$ and $c = 4$.

Checking

Checking whether a point lies on the line or not by substituting given vectors in (1):

$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \mathbf{x}_4 = \begin{pmatrix} \sqrt{2} \\ 4\sqrt{2} \end{pmatrix}, \mathbf{x}_5 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{2}$$

$$\mathbf{n}^\top \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 \end{pmatrix} = \begin{pmatrix} c_1 & c_2 & c_3 & c_4 & c_5 \end{pmatrix} \tag{3}$$

$$\begin{pmatrix} 1 & -2 \end{pmatrix} \begin{pmatrix} 0 & 2 & 4 & \sqrt{2} & 1 \\ 2 & 0 & 0 & 4\sqrt{2} & 1 \end{pmatrix} = \begin{pmatrix} -4 & 2 & 4 & -7\sqrt{2} & -1 \end{pmatrix} \tag{4}$$

$$\tag{5}$$

Conclusion

Conclusion:

The point which lies on the line is only option (3).

C Code

```c
#include<stdio.h>
#include<math.h>
void give_data(double *points){
    points[0] = 1; //Ax
    points[1] = -2; //Ay
    points[2] = 0; //Bx
    points[3] = 2; //By
    points[4] = 2; //Cx
    points[5] = 0; //Cy
    points[6] = 4; //Dx
    points[7] = 0; //Dy
    points[8] = sqrt(2); //EX
    points[9] = 4*sqrt(2); //Ey
    points[10] = 1; //Fx
    points[11] = 1; //Fy
}
```

C Code

```c
double dotpro(double A[], double B[]){
    double sum = 0;
    for(int i = 0; i<2; i++){
        sum += (A[i]*B[i]);
    }
    return sum;
}
int main(){
    double n[2] = {1, -2};
    double A[2] = {0, 2};
    double B[2] = {2, 0};
    double C[2] = {4, 0};
    double D[2] = {sqrt(2), 4*sqrt(2)};
    double E[2] = {1, 1};
    int k = 0;
```

C Code

```c
for(int i = 1; i<=5; i++){
        switch (i){case 1: k = dotpro(n, A); if(k==4){printf("
            Option (%d) lies on the given line.", i);}
            else{printf("Option (%d) does not lie on the given
                line", i);}
            break;
            case 2: k = dotpro(n, B); if(k==4){printf("Option (%d)
                 lies on the given line.", i);}
            else{printf("Option (%d) does not lie on the given
                line", i);}
            break;
```

C Code

```c
            case 3: k = dotpro(n, C); if(k==4){printf("Option (%d)
                 lies on the given line.", i);}
            else{printf("Option (%d) does not lie on the given
                line", i);}
            break;
            case 4: k = dotpro(n, D); if(k==4){printf("Option (%d)
                 lies on the given line.", i);}
            else{printf("Option (%d) does not lie on the given
                line", i);}
            break;
            case 5: k = dotpro(n, E); if(k==4){printf("Option (%d)
                 lies on the given line.", i);}
            else{printf("Option (%d) does not lie on the given
                line", i);}
            break;}}}
```

Python Code 1

```python
import ctypes as ct

lib = ct.CDLL("./problem.so")

lib.give_data.argtypes = [ct.POINTER(ct.c_double)]

points = ct.c_double*12

data = points()

lib.give_data(data)

def send_data():
    return data[0], data[1], data[2], data[3], data[4], data[5],
         data[6], data[7], data[8], data[9]
```

Python Code 2

```python
import matplotlib.pyplot as plt
import numpy as np
import math
from call import send_data
Ax, Ay, Bx, By, Cx, Cy, Dx, Dy, Ex, Ey = send_data()
x = np.linspace(-6, 10, 100)
y = x/2 - 2
X = [Ax, Bx, Cx, Ex, Dy]
Y = [Ay, By, Cy, Ey, Dy]
plt.plot(x, y, 'r-', label="x-2y=4")
plt.plot(X, Y, 'ko')
```

Python Code 2

```
plt.text(8.17, 1.76, "x-2y=4", fontsize=12, color='black')

for i in range(len(X)-1):
    plt.text(X[i]+0.1, Y[i]+0.1, f"({X[i]:.1f},{Y[i]:.1f})",
        fontsize=10, color='black')

plt.text(X[4]+0.1, Y[4]+0.1, f"({X[4]:.1f},{Y[4]:.1f})", fontsize
    =10, color='black')

plt.axvline(x=0, color='k', linewidth=1.5)
```

Python Code 2

```python
plt.axhline(y=0, color='k', linewidth=1.5)
plt.title("Plot of the given line and points")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.axis('equal')
plt.grid(True)
plt.savefig("../figs/plot.png")
plt.show()
```

Plot

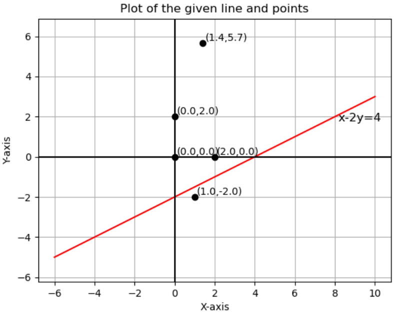

Figure: Plot of given line and points