4.7.56

Vaishnavi - EE25BTECH11059

September 29, 2025

Question

Find the equation of the line whose perpendicular distance from the origin is 4 units and the angle which the normal makes with positive direction of x-axis is 15°

Solution

Variable	Value
d	4
m	$-2-\sqrt{3}$

Table: Variables Used

Solution

Let eq of line be

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

where,

$$\mathbf{n} = \begin{pmatrix} -m \\ 1 \end{pmatrix} \tag{2}$$

$$\mathbf{n} = \begin{pmatrix} 2 + \sqrt{3} \\ 1 \end{pmatrix} \tag{3}$$

Hence eq of line is

$$\begin{pmatrix} 2+\sqrt{3} & 1 \end{pmatrix} \mathbf{x} = c \tag{4}$$

$$\tag{5}$$

As distance from origin=4 units

$$\frac{|c|}{\|n\|} = 4 \tag{6}$$

$$\frac{|c|}{2\sqrt{2+\sqrt{3}}} = 4 \tag{7}$$

$$c = \pm 8\sqrt{2+\sqrt{3}} \tag{8}$$

Hence eq of line is

$$\begin{pmatrix} 2+\sqrt{3} & 1 \end{pmatrix} \mathbf{x} = \pm 8\sqrt{2+\sqrt{3}} \tag{9}$$

Graph

Refer to Figure

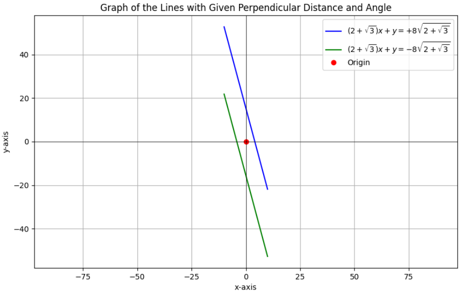

Figure:

Python Code

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

# Define slope and intercept
m = 1/2
c = -3/2

# Create x values
x = np.linspace(-2, 10, 400)

# Equation of the line
y = m * x + c

# Plot the line
plt.figure(figsize=(8,6))
plt.plot(x, y, label= y = (1/2)x - 3/2 , color= blue )

# Mark the y-intercept point
plt.scatter(0, c, color= blue , marker= o , label= y-
```

Python Code

```
# Draw x and y axes
plt.axhline(0, color='black', linewidth=0.8)
plt.axvline(0, color='black', linewidth=0.8)

# Labels and title
plt.xlabel( x-axis )
plt.ylabel( y-axis )
plt.title( Graph of Line: y = (1/2)x - 3/2 )
plt.legend()
plt.grid(True)

# Save the graph as PNG
plt.savefig( grapha , dpi=300)

# Show plot
plt.show()
```

Python Code

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

# Define the line: y = (1/2)x - 2
x = np.linspace(-2, 8, 200)
y = 0.5 * x - 2

# Create plot
plt.figure(figsize=(6,6))
plt.plot(x, y, label=r $y=\frac{1}{2}x-2$ , color= blue )

# Mark intercepts
plt.scatter([4], [0], color= red , label= x-intercept (4,0) , zorder=5)
plt.scatter([0], [-2], color= green , label= y-intercept (0,-2) , zorder=5)

# Axes
```

Python Code

```
# Labels
plt.xlabel( x )
plt.ylabel( y )
plt.title( Line: x - 2y = 4 )
plt.legend()
plt.grid(True)

# Save and show
file_path =  line_plot.png     # will save in current directory
plt.savefig(file_path)
plt.show()

print(f Plot saved as {file_path} )
```

C Code

```c
#include <stdio.h>
#include <math.h>

// Function to compute normal vector, magnitude, and
    constants
void compute_line_params(double m, double d, double* A
    , double* B, double* C1, double* C2) {
    // Normal vector n = (-m, 1)
    *A = -m;
    *B = 1.0;

    // Norm of the vector
    double norm = sqrt((*A) * (*A) + (*B) * (*B));

    // |c| = d * ||n||
    double abs_c = d * norm;

    //  c  values
    *C1 = abs_c;
```

C Code

```
int main() {
    double m = -2.0 - sqrt(3.0);  // Given slope
    double d = 4.0;               // Perpendicular distance from origin

    double A, B, C1, C2;

    // Compute parameters
    compute_line_params(m, d, &A, &B, &C1, &C2);

    // Display the results
    printf( Equation of the lines:\n );
    printf( %.4fx + %.4fy = %.4f\n , A, B, C1);
    printf( %.4fx + %.4fy = %.4f\n , A, B, C2);

    return 0;
}
```

Python and C Code

```
import ctypes
from ctypes import c_double, POINTER

# Load the compiled shared object
lib = ctypes.CDLL( ./code.so )

# Define argument and return types for the function
lib.compute_line_params.argtypes = [c_double, c_double,
                                    POINTER(c_double), POINTER(c_double),
                                    POINTER(c_double), POINTER(c_double)]

# Inputs
m = -2 - 3**0.5  # given slope
d = 4.0          # distance from origin
```

Python and C Code

```
# Outputs
A = c_double()
B = c_double()
C1 = c_double()
C2 = c_double()

# Call the C function
lib.compute_line_params(m, d, ctypes.byref(A), ctypes.
    byref(B), ctypes.byref(C1), ctypes.byref(C2))

# Print the result
print( Equation of the line(s): )
print(f {A.value:.4f}x + {B.value:.4f}y = {C1.value:.4
    f} )
print(f {A.value:.4f}x + {B.value:.4f}y = {C2.value:.4
    f} )
```