4.3.49

Vaishnavi - EE25BTECH11059

September 22, 2025

Question

Write the equation of the lines for which $\tan\theta = \frac{1}{2}$, where θ is the inclination of the line, and

(a) y intercept $-\frac{3}{2}$

(b) x intercept 4

Solution

Variable	**Value**
A	$(0, -\frac{3}{2})$
m	$\frac{1}{2}$

Table: Variables Used

Solution

$$\mathbf{A} = \begin{pmatrix} 0 \\ -\frac{3}{2} \end{pmatrix} \tag{1}$$

$$\text{Let } \mathbf{M} = \begin{pmatrix} 1 \\ m \end{pmatrix} \tag{2}$$

$$\mathbf{M} = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix} \tag{3}$$

Let eq of line be

$$\mathbf{n}^{\mathbf{T}}(\mathbf{x} - \mathbf{A}) = 0 \tag{4}$$

where,

$$\mathbf{n}^{\mathbf{T}}\mathbf{M} = 0 \tag{5}$$

$$\mathbf{n} = \begin{pmatrix} -m \\ 1 \end{pmatrix} \tag{6}$$

$$\mathbf{n} = \begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix} \tag{7}$$

Solution

Hence eq of line is

$$\begin{pmatrix} -\frac{1}{2} & 1 \end{pmatrix} \left(\mathbf{x} - \begin{pmatrix} 0 \\ -\frac{3}{2} \end{pmatrix} \right) = 0 \tag{8}$$

$$\begin{pmatrix} -\frac{1}{2} & 1 \end{pmatrix} \mathbf{x} = -\frac{3}{2} \tag{9}$$

Graph

Refer to Figure

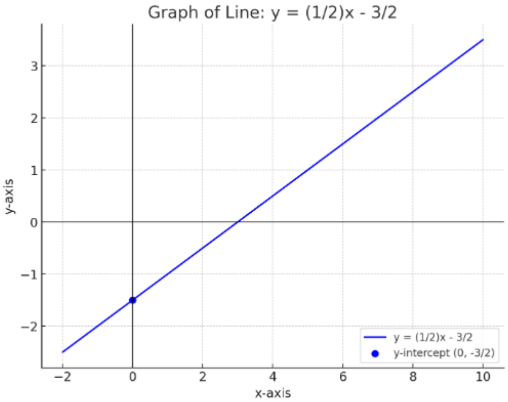


Figure:

Question

Write the equation of the lines for which $\tan\theta = \frac{1}{2}$, where θ is the inclination of the line, and

(a) y intercept $-\frac{3}{2}$

(b) x intercept 4

Solution

Variable	Value
A	$(4, 0)$
m	$\frac{1}{2}$

Table: Variables Used

Solution

$$\mathbf{A} = \begin{pmatrix} 4 \\ 0 \end{pmatrix} \tag{10}$$

$$\text{Let } \mathbf{M} = \begin{pmatrix} 1 \\ m \end{pmatrix} \tag{11}$$

$$\mathbf{M} = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix} \tag{12}$$

solution

Let eq of line be

$$\mathbf{n}^{\mathbf{T}}(\mathbf{x} - \mathbf{A}) = 0 \tag{13}$$

where,

$$\mathbf{n}^{\mathbf{T}}\mathbf{M} = 0 \tag{14}$$

$$\mathbf{n} = \begin{pmatrix} -m \\ 1 \end{pmatrix} \tag{15}$$

$$\mathbf{n} = \begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix} \tag{16}$$

Solution

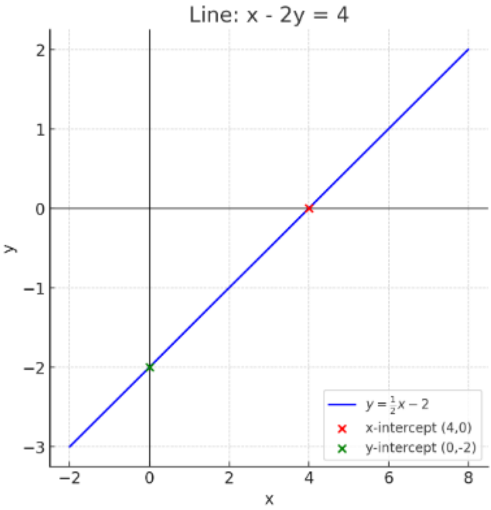
Hence eq of line is

$$\begin{pmatrix} -\frac{1}{2} & 1 \end{pmatrix} (\mathbf{x} - \begin{pmatrix} 4 \\ 0 \end{pmatrix}) = 0 \tag{17}$$

$$\begin{pmatrix} -\frac{1}{2} & 1 \end{pmatrix} \mathbf{x} = -2 \tag{18}$$

Graph

Refer to Figure



Python Code 1

```
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 1

```
# 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 2

```
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 2

```
# 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

```
#include <stdio.h>

// Print 2x3 matrix
void printMatrix(float mat[2][3]) {
    for(int i=0; i<2; i++) {
        for(int j=0; j<3; j++) {
            printf( %6.2f  , mat[i][j]);
        }
        printf( \n );
    }
    printf( \n );
}

// Gaussian Elimination
void gaussElimination(float mat[2][3]) {
    // Normalize first row
    float factor = mat[0][0];
    if (factor != 0) {
        for(int j=0; j<3; j++)
```

C Code

```c
    // Eliminate below
     factor = mat[1][0];
     for(int j=0; j<3; j++)
         mat[1][j] -= factor * mat[0][j];

     // Normalize second row
     factor = mat[1][1];
     if (factor != 0) {
         for(int j=0; j<3; j++)
             mat[1][j] /= factor;
     }

     // Eliminate above
     factor = mat[0][1];
     for(int j=0; j<3; j++)
         mat[0][j] -= factor * mat[1][j];
}
```

Python and C Code

```
import ctypes
import numpy as np

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

# Define function signatures
lib.gaussElimination.argtypes = [((ctypes.c_float * 3)
    * 2)]
lib.printMatrix.argtypes = [((ctypes.c_float * 3) * 2)
   ]

def to_c_matrix(py_mat):
        Convert Python 2x3 list into C float[2][3]
    c_mat = ((ctypes.c_float * 3) * 2)()
    for i in range(2):
        for j in range(3):
            c_mat[i][j] = py_mat[i][j]
    return c_mat
```

Python and C Code

```
def to_py_matrix(c_mat):
        Convert C float[2][3] back to Python list
    return [[c_mat[i][j] for j in range(3)] for i in
        range(2)]

# Case (a): slope=1/2, y-intercept=-3/2
mat1 = [[0, 1, -1.5],
         [1, -0.5, 0]]

c_mat1 = to_c_matrix(mat1)
print( Case (a): before elimination: )
lib.printMatrix(c_mat1)

lib.gaussElimination(c_mat1)
```

Python and C Code

```
print( Case (a): after elimination: )
lib.printMatrix(c_mat1)
print( Python Matrix: , to_py_matrix(c_mat1))

# Case (b): slope=1/2, x-intercept=4
mat2 = [[4, 1, 0],
        [1, -0.5, 0]]

c_mat2 = to_c_matrix(mat2)
print( \nCase (b): before elimination: )
lib.printMatrix(c_mat2)

lib.gaussElimination(c_mat2)

print( Case (b): after elimination: )
lib.printMatrix(c_mat2)
print( Python Matrix: , to_py_matrix(c_mat2))
```