4.7.59

Vivek K Kumar - EE25BTECH11062

September 30, 2025

Question

Find the equation of a line perpendicular to the line $x + 2y + 3 = 0$ and passing through the point $(1, -2)$.

Variables used

(1)

Point	Value
$\mathbf{n_1}$	$\begin{pmatrix} 1 \\ 2 \end{pmatrix}$
$\mathbf{n_2}$	$\begin{pmatrix} 1 \\ k \end{pmatrix}$
c	-3
A	$\begin{pmatrix} 1 \\ -2 \end{pmatrix}$

Table: Variables used

Solution

The given line can be expressed as

$$\mathbf{n_1}^\top \mathbf{x} = c \tag{2}$$

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

$$\tag{4}$$

As the given lines are perpendicular

$$\mathbf{n_1}^\top \mathbf{n_2} = 0 \tag{5}$$

$$k = \frac{-1}{2} \tag{6}$$

$$\mathbf{n_2} = \begin{pmatrix} 1 \\ -1/2 \end{pmatrix} \tag{7}$$

Solution

The equation of the resulting line can be expressed as

$$\mathbf{n_2}^\top (\mathbf{x} - \mathbf{A}) = 0 \tag{8}$$

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

$$\begin{pmatrix} 1 & \frac{-1}{2} \end{pmatrix} \mathbf{x} = 2 \tag{10}$$

Python - Importing libraries and checking system

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

from libs.line.funcs import *
from libs.triangle.funcs import *
from libs.conics.funcs import circ_gen

import subprocess
import shlex

print('Using termux?(y/n)')
y = input()
```

Python - Writing coordinates of point and corresponding line direction vectors

```
x = np.array([1, -2]).reshape(-1,1)
m1 = np.array([-2, 1]).reshape(-1,1)
m2 = np.array([1/2, 1]).reshape(-1,1)
```

Python - Generating points and plotting

```
p_l1 = line_gen(x-5*m1, x+5*m1)
p_l2 = line_gen(x-5*m2, x+5*m2)
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(p_l1[0, :], p_l1[1, :], label = 'Given line')
ax.plot(p_l2[0, :], p_l2[1, :], label = 'Required Line')
```

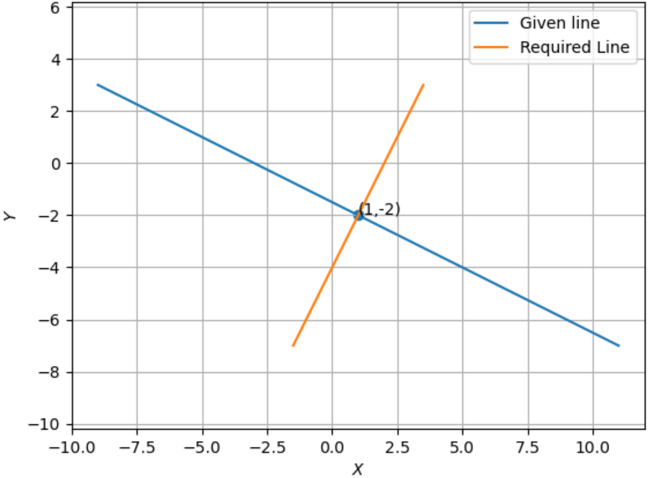
Python - Labelling points

```
ax.scatter(np.array([x[0]]), np.array([x[1]]))
ax.text(x[0], x[1], s='(1,-2)')

ax.set_xlabel('$X$')
ax.set_ylabel('$Y$')
ax.legend(loc='best')
ax.grid(True)
ax.axis('equal')
```

Python - Saving figure and opening it

```python
fig.savefig('../figs/fig.png')
print('Saved figure to ../figs/fig.png')

if(y == 'y'):
    subprocess.run(shlex.split('termux-open ../figs/fig.png'))
else:
    subprocess.run(["open", "../figs/fig.png"])
```

Plot-Using only Python



C Code (0) - Importing libraries

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include "libs/matfun.h"
#include "libs/geofun.h"
```

C Code (1) - Function to Generate Points on a Line

```c
void point_gen(FILE *p_file, double **A, double **B, int rows,
    int cols, int npts){
    for(int i = 0; i <= npts; i++){
     double **output = Matadd(A, Matscale(Matsub(B, A, rows, cols
         ), rows, cols, (double)i/npts), rows, cols);
     fprintf(p_file, "%lf, %lf\n", output[0][0], output[1][0]);
     freeMat(output, rows);
    }
}
```

C Code (2) - Function to write points of a line by using the given points to a file

```c
void write_points(double x1, double y1, double x2, double y2,
    double x3, double y3, int npts){
    int m = 2;
    int n = 1;

    double **A = createMat(m, n);
    double **B = createMat(m, n);
    double **C = createMat(m, n);

    B[0][0] = x2;
    B[1][0] = y2;
```

C Code (2) - Function to write points of a line by using the given points to a file

```c
    A[0][0] = x1;
    A[1][0] = y1;

    C[0][0] = x3;
    C[1][0] = y3;

    double **L1_1 = Matsub(A, Matscale(B, m, n, -5), m, n);
    double **L1_2 = Matsub(A, Matscale(B, m, n, 5), m, n);
    double **L2_1 = Matsub(A, Matscale(C, m, n, -5), m, n);
    double **L2_2 = Matsub(A, Matscale(C, m, n, 5), m, n);

    FILE *p_file;
    p_file = fopen("plot.dat", "w");

    if(p_file == NULL)
        printf("Error opening one of the data files\n");
```

C Code (2) - Function to write points of a line by using the given points to a file

```
    point_gen(p_file, L1_1, L1_2, m, n, npts);
    point_gen(p_file, L2_1, L2_2, m, n, npts);

    freeMat(A, m);
    freeMat(B, m);
    freeMat(C, m);
    freeMat(L1_1, m);
    freeMat(L1_2, m);
    freeMat(L2_1, m);
    freeMat(L2_2, m);

    fclose(p_file);
}
```

Python Code (0) - Importing libraries and checking system

```python
import numpy as np
import matplotlib.pyplot as plt
import ctypes
import os
import sys
import subprocess
import math

print('Using termux? (y/n)')
termux = input()
```

Python Code (1) - Using Shared Object

```
lib_path = os.path.join(os.path.dirname(__file__), 'plot.so')
my_lib = ctypes.CDLL(lib_path)

my_lib.write_points.argtypes = [ctypes.c_double, ctypes.c_double,
     ctypes.c_double, ctypes.c_double, ctypes.c_double, ctypes.
    c_double, ctypes.c_int]
my_lib.write_points.restype = None
x = np.array([1, -2]).reshape(-1, 1)
m1 = np.array([-2, 1]).reshape(-1, 1)
m2 = np.array([1/2, 1]).reshape(-1, 1)
npts = 20000
```

Python Code (2) - Loading points and plotting them

```
my_lib.write_points(x[0][0], x[1][0], m1[0][0], m1[1][0], m2
    [0][0], m2[1][0], npts)

fig = plt.figure()
ax = fig.add_subplot(111)
labels = ['Given Line', 'Required Line']
pts = np.block([x])

for i,label in enumerate(labels):
    points = np.loadtxt('plot.dat', delimiter = ',', usecols
        =(0,1))[i*(npts+1):(i+1)*(npts+1)]
    ax.plot(points[:, 0], points[:, 1], label = label)

ax.text(pts[:, 0][0], pts[:, 0][1], s=f'(1, -2)')
```

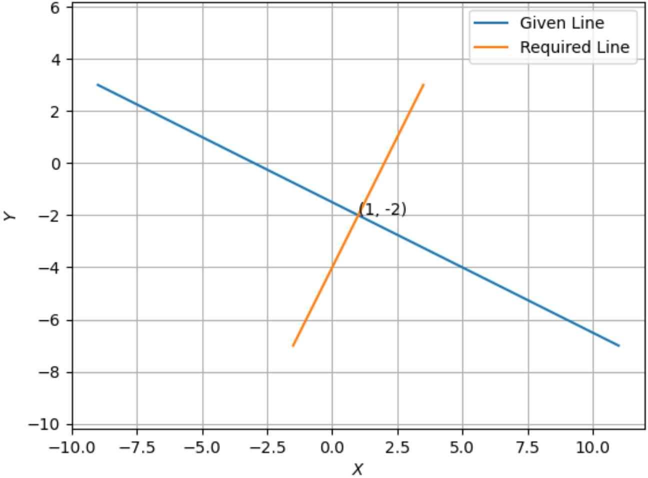
Python Code (3) - Labelling plot

```python
ax.set_xlabel('$X$')
ax.set_ylabel('$Y$')
ax.legend(loc='best')
ax.grid()
ax.axis('equal')
```

Python Code (4) - Saving and displaying plot

```python
fig.savefig('../figs/fig2.png')
print('Saved figure to ../figs/fig2.png')

if(termux == 'y'):
    subprocess.run(shlex.split('termux-open ../figs/fig2.png'))
else:
    subprocess.run(["open", "../figs/fig2.png"])
```

Plot-Using Both C and Python


Given Line
Required Line
(1, -2)
Y
X