

4.13.66

Vivek K Kumar - EE25BTECH11062

October 10, 2025

Question

A straight line L with negative slope passes through the point $(8, 2)$ and cuts the positive coordinate axes at points P and Q . Find the absolute minimum value of $OP + OQ$, as L varies, where O is the origin.

Variables used

(1)

Point	Value
m	$\begin{pmatrix} 1 \\ -m \end{pmatrix} \quad m > 0$
h	$\begin{pmatrix} 8 \\ 2 \end{pmatrix}$

Table: Variables used

Solution

It is known that

$$\mathbf{e}_1^\top \mathbf{P} = 0 \quad (2)$$

$$\mathbf{e}_2^\top \mathbf{Q} = 0 \quad (3)$$

Given line L can be represented as

$$\mathbf{x} = \mathbf{h} + k\mathbf{m} \quad (4)$$

$$\mathbf{e}_1^\top \mathbf{P} = \mathbf{e}_1^\top \mathbf{h} + k_1 \mathbf{e}_1^\top \mathbf{m} \quad (5)$$

$$k_1 = -\frac{\mathbf{e}_1^\top \mathbf{h}}{\mathbf{e}_1^\top \mathbf{m}} \quad (6)$$

$$\mathbf{P} = \mathbf{h} - \frac{\mathbf{e}_1^\top \mathbf{h}}{\mathbf{e}_1^\top \mathbf{m}} \mathbf{m} \quad (7)$$

$$\mathbf{e}_2^\top \mathbf{Q} = \mathbf{e}_2^\top \mathbf{h} + k_2 \mathbf{e}_2^\top \mathbf{m} \quad (8)$$

$$k_2 = -\frac{\mathbf{e}_2^\top \mathbf{h}}{\mathbf{e}_2^\top \mathbf{m}} \quad (9)$$

$$\mathbf{Q} = \mathbf{h} - \frac{\mathbf{e}_2^\top \mathbf{h}}{\mathbf{e}_2^\top \mathbf{m}} \mathbf{m} \quad (10)$$

$$(11)$$

Substituting values

$$\mathbf{P} = \begin{pmatrix} 8 \\ 2 \end{pmatrix} - 8 \begin{pmatrix} 1 \\ -m \end{pmatrix} \quad (12)$$

$$= \begin{pmatrix} 0 \\ 2 + 8m \end{pmatrix} \quad (13)$$

$$\mathbf{Q} = \begin{pmatrix} 8 \\ 2 \end{pmatrix} - \frac{2}{m} \begin{pmatrix} 1 \\ -m \end{pmatrix} \quad (14)$$

$$= \begin{pmatrix} 8 + \frac{2}{m} \\ 0 \end{pmatrix} \quad (15)$$

Solution

We have to find the minimum of $\|\mathbf{P}\| + \|\mathbf{Q}\|$

$$\|\mathbf{P}\| + \|\mathbf{Q}\| = 2 + 8m + 8 + \frac{2}{m} \quad (16)$$

$$= 10 + 8m + \frac{2}{m} \quad (17)$$

Applying AM-GM inequality

$$\frac{8m + \frac{2}{m}}{2} \geq \sqrt{8m \cdot \frac{2}{m}} \quad (18)$$

$$\geq 4 \quad (19)$$

$$\implies 10 + 8m + \frac{2}{m} \geq 18 \quad (20)$$

Hence we can write

$$\|\mathbf{P}\| + \|\mathbf{Q}\| \geq 18 \quad (21)$$

Hence, $\min(\|\mathbf{P}\| + \|\mathbf{Q}\|) = 18$

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 - Setting up Variables and plotting the line equation

```
A = np.array([8, 2]).reshape(-1,1)
m = np.array([1, -1/2]).reshape(-1,1)

p_l1 = line_gen(A-8*m, A+4*m)

plt.plot(p_l1[0, :], p_l1[1, :], label = 'Line with min OP+OQ')

pts = np.block([A-8*m, A+4*m, A])
labels = ['P(0,6)', 'Q(12,0)', 'A(8,2)']
plt.scatter(pts[0, :], pts[1, :])
for i, txt in enumerate(labels):
    plt.annotate(txt, (pts[0, i], pts[1, i]), textcoords="
        offset points", xytext=(20,5), ha='center')
```

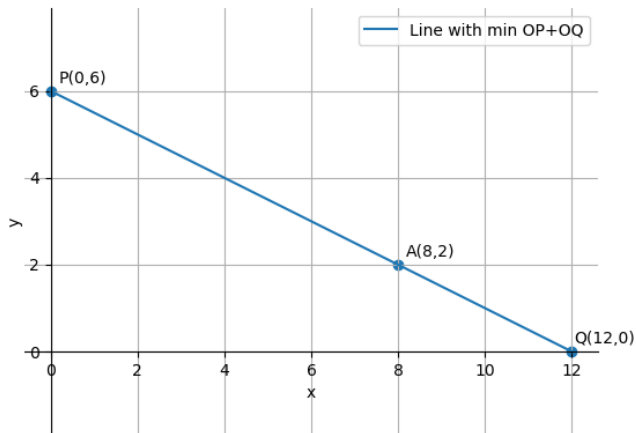
Python - Setting up plot and labeling axes

```
ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['left'].set_position('zero')
plt.xlabel('x')
plt.ylabel('y')
plt.legend(loc='best')
plt.grid()
plt.axis('equal')
```

Python - Saving figure and opening it

```
1 fig.savefig('../figs/fig.png')
2 print('Saved figure to ../figs/fig.png')
3
4 if(y == 'y'):
5     subprocess.run(shlex.split('termux-open ../figs/fig.png'))
6 else:
7     subprocess.run(["open", "../figs/fig.png"])
```

Plot-Using only Python



C Code (0) - Importing libraries

```
#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

```
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 b/w given points to a file

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

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

    A[0][0] = x1-8*x2;
    A[1][0] = y1-8*y2;
```

C Code (2) - Function to write points b/w given 2 points to a file

```
B[0][0] = x1+4*x2;
B[1][0] = y1+4*y2;

FILE *p_file;
p_file = fopen("plot.dat", "w");
if(p_file == NULL)
    printf("Error opening one of the data files\n");

point_gen(p_file, A, B, m, n, npts);

freeMat(A, m);
freeMat(B, m);

fclose(p_file);
}
```


Python Code (0) - Importing libraries and checking system

```
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_int]
my_lib.write_points.restype = None
A = np.array([8, 2]).reshape(-1, 1)
m = np.array([1, -1/2]).reshape(-1, 1)
npts = 20000
```

Python Code (2) - Loading points and plotting them

```
my_lib.write_points(A[0][0], A[1][0], m[0][0], m[1][0], npts)

labels = ['Line with min  $OP+OQ$ ']
point_labels = ['P(0, 6)', 'Q(12, 0)', 'A(8,2)']
pts = np.block([A-8*m, A+4*m, A])

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

for i,label in enumerate(point_labels):
    plt.annotate(label, (pts[0,i],pts[1,i]), textcoords="offset
        points", xytext=(20,5),ha='center')
```

Python Code (3) - Setting up plot and labeling axes

```
ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['left'].set_position('zero')
plt.xlabel('x')
plt.ylabel('y')
plt.legend(loc='best')
plt.grid()
plt.axis('equal')
```

Python Code (4) - Saving and displaying plot

```
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

