

2.9.13

Vivek K Kumar - EE25BTECH11062

September 13, 2025

Question

If the two lines If **A**, **B** and **C** are the position vectors of the points **A** (2, 3, -4), **B** (3, -4, -5) and **C** (3, 2, -3) respectively, then $\|\mathbf{A} + \mathbf{B} + \mathbf{C}\|$ is equal to:

Variables used

Name	Point
A	$\begin{pmatrix} 2 \\ 3 \\ -4 \end{pmatrix}$
B	$\begin{pmatrix} 3 \\ -4 \\ -5 \end{pmatrix}$
C	$\begin{pmatrix} 3 \\ 2 \\ -3 \end{pmatrix}$

Table: Variables Used

Solution

Finding $\mathbf{A} + \mathbf{B} + \mathbf{C}$,

$$\mathbf{A} + \mathbf{B} + \mathbf{C} = \begin{pmatrix} 2 \\ 3 \\ -4 \end{pmatrix} + \begin{pmatrix} 3 \\ -4 \\ -5 \end{pmatrix} + \begin{pmatrix} 3 \\ 2 \\ -3 \end{pmatrix} \quad (1)$$

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

Now finding $\|\mathbf{A} + \mathbf{B} + \mathbf{C}\|$,

$$\|\mathbf{A} + \mathbf{B} + \mathbf{C}\|^2 = (\mathbf{A} + \mathbf{B} + \mathbf{C})^\top (\mathbf{A} + \mathbf{B} + \mathbf{C}) \quad (3)$$

$$= \begin{pmatrix} 8 & 1 & -12 \end{pmatrix} \begin{pmatrix} 8 \\ 1 \\ -12 \end{pmatrix} \quad (4)$$

Solution

$$= 8^2 + 1^2 + (-12)^2 \quad (5)$$

$$= 209 \quad (6)$$

Hence,

$$\|\mathbf{A} + \mathbf{B} + \mathbf{C}\| = \sqrt{209} \quad (7)$$

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

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 - Finding norm of the sum of vectors

```
A = np.array([2, 3, -4]).reshape(-1, 1)
B = np.array([3, -4, -5]).reshape(-1, 1)
C = np.array([3, 2, -3]).reshape(-1, 1)
O = np.zeros(3).reshape(-1,1)

P = A+B+C

norm = LA.norm(P)

print(f'Norm of A+B+C = {norm}')
```

Python - Generating points and plotting

```
p_A = line_gen(0, A)
p_B = line_gen(0, B)
p_C = line_gen(0, C)
p_P = line_gen(0, P)

fig = plt.figure()
ax = fig.add_subplot(111, projection = '3d')
ax.plot(p_A[0, :], p_A[1, :], p_A[2, :], label = 'Vector A')
ax.plot(p_B[0, :], p_B[1, :], p_B[2, :], label = 'Vector B')
ax.plot(p_C[0, :], p_C[1, :], p_C[2, :], label = 'Vector C')
ax.plot(p_P[0, :], p_P[1, :], p_P[2, :], label = 'Vector A+B+C')
```


Python - Labelling points

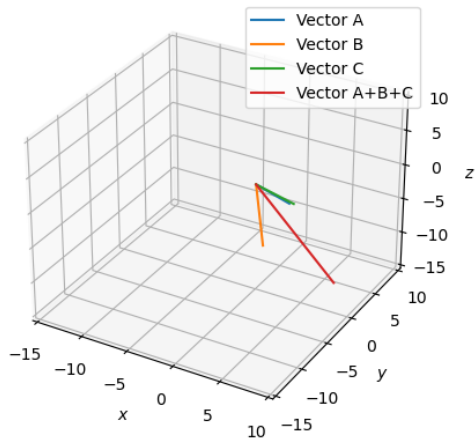
```
1 ax.set_xlabel('$x$')
2 ax.set_ylabel('$y$')
3 ax.set_zlabel('$z$')
4 ax.legend(loc='best')
5 ax.grid(True)
6 ax.axis('equal')
7 ax.set_xlim([-15, 10])
8 ax.set_ylim([-15, 10])
9 ax.set_zlim([-15, 10])
```

Python - Saving figure and opening it

```
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

```
#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, %lf\n", output[0][0], output
            [1][0], output[2][0]);
        freeMat(output, rows);
    }
}
```

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

```
float find_norm(double **p, int n);

int write_points(double x1, double y1, double z1, double x2,
    double y2, double z2, double x3, double y3, double z3, int
    npts){
    int m = 3;
    int n = 1;

    double **R = createMat(m, n);
    double **O = createMat(m, n);
    double **T = createMat(m, n);
    double **S = createMat(m, n);
    double **origin = createMat(m, n);

    for(int i = 0; i<3; i++)
        origin[i][0] = 0;
```

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

```
R[0][0] = x2;
```

```
R[1][0] = y2;
```

```
R[2][0] = z2;
```

```
O[0][0] = x1;
```

```
O[1][0] = y1;
```

```
O[2][0] = z1;
```

```
T[0][0] = x1+x2+x3;
```

```
T[1][0] = y1+y2+y3;
```

```
T[2][0] = z1+z2+z3;
```

```
S[0][0] = x3;
```

```
S[1][0] = y3;
```

```
S[2][0] = z3;
```

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

```
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, origin, O, m, n, npts);
point_gen(p_file, origin, R, m, n, npts);
point_gen(p_file, origin, S, m, n, npts);
point_gen(p_file, origin, T, m, n, npts);
float k = find_norm(T, m);
freeMat(R, m);
freeMat(O, m);
freeMat(T, m);
freeMat(S, m);
freeMat(origin, m);
fclose(p_file);
return k;
}
```


C Code (3) - Finding norm

```
float find_norm(double **p, int n){  
    return sqrt(Matmul(transposeMat(p, n, 1), p, 1, n, 1)[0][0]);  
}
```

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

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_double, ctypes.c_double,
                                ctypes.c_int]
my_lib.write_points.restype = ctypes.c_float
p1 = np.array([2, 3, -4])
p2 = np.array([3, -4, -5])
p3 = np.array([3, 2, -3])
npts = 20000
k = my_lib.write_points(p1[0], p1[1], p1[2], p2[0], p2[1], p2[2],
                        p3[0], p3[1], p3[2], npts)
```

Python Code (2) - Loading points and finding norm

```
print(f'The norm of A+B+C = {k}')
```



```
points = np.loadtxt('plot.dat', delimiter=',', usecols = (0,1, 2)
    )[0:npts+1]
```



```
points2 = np.loadtxt('plot.dat', delimiter=',', usecols = (0,1,
    2))[npts+1:2*(npts+1)]
```



```
points3 = np.loadtxt('plot.dat', delimiter = ',', usecols =
    (0,1,2))[2*(npts+1):3*(npts+1)]
```



```
points4 = np.loadtxt('plot.dat', delimiter = ',', usecols =
    (0,1,2))[3*(npts+1):4*(npts+1)]
```

Python Code (2) - Loading points and finding norm

```
x = points[:, 0]
y = points[:, 1]
z = points[:, 2]

x2 = points2[:, 0]
y2 = points2[:, 1]
z2 = points2[:, 2]

x3 = points3[:, 0]
y3 = points3[:, 1]
z3 = points3[:, 2]

x4 = points4[:, 0]
y4 = points4[:, 1]
z4 = points4[:, 2]
```

Python Code (3) - Plotting points

```
fig = plt.figure()
ax = fig.add_subplot(111, projection = '3d')
ax.plot(x, y, z, label = 'Vector A')
ax.plot(x2, y2, z2, label = 'Vector B')
ax.plot(x3, y3, z3, label = 'Vector C')
ax.plot(x4, y4, z4, label = 'Vector A+B+C')

ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
ax.set_zlabel('$z$')
ax.legend(loc='best')
ax.grid()
ax.axis('equal')
ax.set_xlim([-15, 10])
ax.set_ylim([-15, 10])
ax.set_zlim([-15, 10])
```

Python Code (4) - Saving plot and opening it

```
1 fig.savefig('../figs/fig2.png')
2 print('Saved figure to ../figs/fig2.png')
3
4 if(termux == 'y'):
5     subprocess.run(shlex.split('termux-open ../figs/fig2.png'))
6 else:
7     subprocess.run(["open", "../figs/fig2.png"])
```

Plot-Using Both C and Python

