2.10.61

Kartik Lahoti - EE25BTECH11032

September 5, 2025

Question

If $\mathbf{a}$ and $\mathbf{b}$ are vectors such that $\left|\mathbf{a}+\mathbf{b}\right| = \sqrt{29}$ and

$$\mathbf{a} \times \left(2\hat{i} + 3\hat{j} + 4\hat{k}\right) = \left(2\hat{i} + 3\hat{j} + 4\hat{k}\right) \times \mathbf{b}$$

then a possible value of $(\mathbf{a}+\mathbf{b}) \cdot \left(-7\hat{i} + 2\hat{j} + 3\hat{k}\right)$ is

(1) 0 (2) 3 (3) 4 (4) 8

Theoretical Solution

Given :

Symbol	Value	Description
c	$\begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}$	Given Point
d	$\begin{pmatrix} -7 \\ 2 \\ 3 \end{pmatrix}$	Given Point
$\mathbf{a}+\mathbf{b}$	?	Desired Point

Table: 2.10.61

Theory

$$\mathbf{a} \times \mathbf{c} = \mathbf{c} \times \mathbf{b} \tag{1}$$

$$\mathbf{a} \times \mathbf{c} = -(\mathbf{b} \times \mathbf{c}) \tag{2}$$

$$(\mathbf{a} + \mathbf{b}) \times \mathbf{c} = \mathbf{0} \tag{3}$$

Theoretical Solution

If cross product of 2 vectors is zero , this implies both the vectors are parallel. Thus ,

$$(\mathbf{a}+\mathbf{b}) \parallel \mathbf{c} \tag{4}$$

$$\therefore (\mathbf{a}+\mathbf{b}) = \lambda\mathbf{c}, \text{ where } \lambda \in \mathbb{R} \tag{5}$$

Theoretical Solution

Equating the magnitudes , we get

$$\|(\mathbf{a}+\mathbf{b})\|^2 = \lambda^2 \|c\|^2 \tag{6}$$

$$29 = \lambda^2 29 \tag{7}$$

$$\lambda = \pm 1 \tag{8}$$

Theoretical Solution

Thus,

$$(\mathbf{a}+\mathbf{b}) = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix} \text{ or } (\mathbf{a}+\mathbf{b}) = \begin{pmatrix} -2 \\ -3 \\ -4 \end{pmatrix} \tag{9}$$

Hence,

$$(\mathbf{a}+\mathbf{b})^{\top}\mathbf{d} = 4 \text{ or } -4 \tag{10}$$

Answer : Option (3)

C Code (1)

```c
#include <math.h>
double norm_vec_sq(double *A , int m )
{
    double sum = 0.0;
    for ( int i = 0 ; i < m ; i++ )
    {
        sum += pow(A[i] , 2 );
    }
    return sum;
}
```

C Code (2) - Function to Generate Points on Line

```c
void linegen(double *X, double *Y , double *Z , double *A ,
    double *B , int n , int m )
{
    double temp[m] ;
    for (int i = 0 ; i < m ; i++)
    {
        temp [ i ] = (B[i]- A[i]) /(double) n ;
    }
    for (int i = 0 ; i <= n ; i++ )
    {
        X[i] = A[0] + temp[0] * i ;
        Y[i] = A[1] + temp[1] * i ;
        Z[i] = A[2] + temp[2] * i ;
    }
}
```

Python Code - Using Shared Object

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt
handc1 = ctypes.CDLL("./func.so")

handc1.norm_vec_sq.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.c_int]

handc1.norm_vec_sq.restype = ctypes.c_double
C = np.array([[2],[3],[4]], dtype= np.float64).reshape(-1,1)
ab_sq = 29
m = 3
c_sq = handc1.norm_vec_sq(
    C.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),m)
```

Python Code - Using Shared Object

```
l = ab_sq / c_sq
ab1 = np.sqrt(l) * C
ab2 = -np.sqrt(l) * C

def line_cre(P: np.ndarray , Q: np.ndarray, str):
    handc2 = ctypes.CDLL("./line_gen.so")

    handc2.linegen.argtypes = [
        ctypes.POINTER(ctypes.c_double),
        ctypes.POINTER(ctypes.c_double),
        ctypes.POINTER(ctypes.c_double),
        ctypes.POINTER(ctypes.c_double),
        ctypes.POINTER(ctypes.c_double),
        ctypes.c_int , ctypes.c_int
    ]
```

Python Code - Using Shared Object

```
handc2.linegen.restype = None
    n = 200
    X_l = np.zeros(n,dtype=np.float64)
    Y_l = np.zeros(n,dtype=np.float64)
    Z_l = np.zeros(n,dtype=np.float64)
    handc2.linegen (
        X_l.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
        Y_l.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
        Z_l.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
        P.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
        Q.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
        n,3
    )
    ax.plot([X_l[0],X_l[-1]],[Y_l[0],Y_l[-1]],[Z_l[0],Z_l[-1]],
        str)
```

Python Code - Using Shared Object

```
O = np.array([[0],[0],[0]]).reshape(-1,1)
fig = plt.figure()
ax = fig.add_subplot(111,projection="3d")

line_cre(ab1,O,"g-")
line_cre(ab2,O,"r-")

coords = np.block([[ab1,ab2,O]])
ax.scatter(coords[0,:],coords[1,:],coords[2,:])
vert_labels = [r'$(a+b)_1$',r'$(a+b)_2$','O']
```

Python Code - Using Shared Object

```
for i, txt in enumerate(vert_labels):
    if (coords[0,i] == 0 ) :
        ax.text(coords[0,i], coords[1,i] , coords[2,i],txt , ha='
            center', va = 'bottom')
    else :
        ax.text(coords[0,i], coords[1,i] , coords[2,i],f'{txt}\n
            ({coords[0,i]:.1f}, {coords[1,i]:.1f}, {coords[2,i]:.1
            f})',ha='center', va = 'bottom')
ax.scatter(coords[0,2], coords[1,2], coords[2,2], color="b",
    label="O : ORIGIN")
```

Python Code - Using Shared Object

```
ax.legend(loc = "best")
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
ax.set_zlabel('$z$')
ax.grid()
plt.title("Fig:2.10.61")
ax.set_box_aspect([1,1,1])

fig.savefig("../figs/vector1.png")
fig.show()

#plt.savefig('figs/triangle/ang-bisect.pdf')
#subprocess.run(shlex.split("termux-open figs/triangle/ang-bisect
    .pdf"))
```

Python Code

```
import math
import sys
sys.path.insert(0, '/home/kartik-lahoti/matgeo/codes/CoordGeo')
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

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


#if using termux
#import subprocess
#import shlex
```

Python Code

```
C = np.array([[2],[3],[4]], dtype = np.float64 ).reshape(-1,1)
ab_sq= 29

c_sq = LA.norm(C)**2

l = ab_sq / c_sq

ab1 = np.sqrt(l) * C
ab2 = - np.sqrt(l) * C
O = np.array([0,0,0]).reshape(-1,1)
```

Python Code

```
def plot_it(P,Q,str):
    x_l = line_gen_num(P,Q,20)
    ax.plot(x_l[0,:],x_l[1,:],x_l[2,:] , str )

fig = plt.figure()
ax = fig.add_subplot(111,projection = "3d")

plot_it(ab1,O,"g-")
plot_it(ab2,O,"r-")

coords = np.block([[ab1,ab2,O]])
plt.scatter(coords[0,:],coords[1,:],coords[2,:])
vert_labels = [r'$(a+b)_1$',r'$(a+b)_2$','O']
```

Python Code

```
for i, txt in enumerate(vert_labels):
    if (coords[0,i] == 0 ) :
        ax.text(coords[0,i], coords[1,i] , coords[2,i],txt , ha='
            center', va = 'bottom')
    else :
        ax.text(coords[0,i], coords[1,i] , coords[2,i],f'{txt}\n
            ({coords[0,i]:.1f}, {coords[1,i]:.1f}, {coords[2,i]:.1
            f})',ha='center', va = 'bottom')

ax.scatter(coords[0,2], coords[1,2], coords[2,2], color="b",
    label="O : ORIGIN")
ax.legend(loc = "best")
```

Python Code

```
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
ax.set_zlabel('$z$')

ax.grid()
plt.title("Fig:2.8.15")
ax.set_box_aspect([1,1,1])

fig.savefig("../figs/vector2.png")
fig.show()
#plt.savefig('figs/triangle/ang-bisect.pdf')
#subprocess.run(shlex.split("termux-open figs/triangle/ang-bisect
    .pdf"))
```

Plot