

1.10.30

Vivek K Kumar - EE25BTECH11062

August 26, 2025

# Question

The direction cosines of the vector  $\begin{pmatrix} 2 \\ 2 \\ -1 \end{pmatrix}$  are \_\_\_\_\_

# Variables used

Name	Point
$\begin{pmatrix} 2 \\ 2 \\ -1 \end{pmatrix}$	Vector $R$

Table: Variables Used

The unit vector along the direction of given vector is

$$\frac{A}{\|A\|} = \frac{1}{3} \begin{pmatrix} 2 \\ 2 \\ -1 \end{pmatrix} \quad (1)$$

$$= \begin{pmatrix} \frac{2}{3} \\ \frac{2}{3} \\ \frac{-1}{3} \end{pmatrix} \quad (2)$$

# 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 direction cosines

```
R = np.array([2, 2, -1]).reshape(-1, 1)
O = np.zeros(3).reshape(-1, 1)
norm_R = LA.norm(R)
X = R/norm_R
print(f"The direction cosines of the given vector is \n{X}")
```

# Python - Generating points and plotting

```
p_OR = line_gen(0, R)
p_OX = line_gen(0, X)

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

ax.plot(p_OR[0, :], p_OR[1, :], p_OR[2, :], label = 'Line through
OR')
ax.plot(p_OX[0, :], p_OX[1, :], p_OX[2, :], label = 'Direction
cosines of OR')
```

# Python - Labelling points

```
line_coords = np.block([[0, R, X]])
ax.scatter(line_coords[0,:], line_coords[1,:], line_coords[2, :])
vert_labels = ['O', 'R', 'X']
for i, txt in enumerate(vert_labels):
    ax.text(line_coords[0][i], line_coords[1][i], line_coords[2][i], txt, color='red')

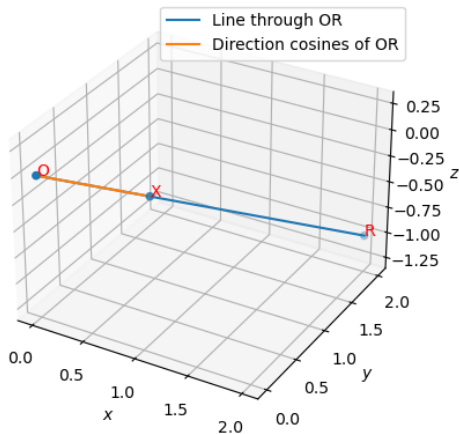
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
ax.set_zlabel('$z$')
ax.legend(loc='best')
ax.grid(True)
ax.axis('equal')
```



# 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, %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

```
void calculate_unit(double **R, int npts);

void write_points(double x1, double y1, double z1, int npts){
    int m = 3;
    int n = 1;

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

    R[0][0] = x1;
    R[1][0] = y1;
    R[2][0] = z1;

    O[0][0] = 0;
    O[1][0] = 0;
    O[2][0] = 0;
```

## 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 data file\n");
}

point_gen(p_file, 0, R, m, n, npts);
calculate_unit(R, npts);

freeMat(R, m);
freeMat(0, m);

fclose(p_file);
}
```

## C Code (3) - Finding unit vector

```
void calculate_unit(double **R, int npts){
    double **X = Matunit(R, 3);
    double **O = createMat(3, 1);

    for(int i = 0; i<3; i++){
        O[i][0] = 0;
    }

    FILE *p_file;
    p_file = fopen("plot2.dat", "w");
    if(p_file == NULL){
        printf("Error opening data file\n");
    }

    point_gen(p_file, O, X, 3, 1, npts);
}
```

## C Code (3) - Finding unit vector

```
1 freeMat(X, 3);  
2 freeMat(O, 3);  
3  
4 fclose(p_file);  
5 }
```



# 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_int]
my_lib.write_points.restype = None
my_lib.write_points(2, 2, -1, 20000)
```

## Python Code (2) - Loading points and finding unit vector

```
points = np.loadtxt('plot.dat', delimiter=',', usecols = (0,1, 2)
    )
points2 = np.loadtxt('plot2.dat', delimiter=',', usecols = (0,1,
    2))

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

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

print(f"The directions cosines of OR are \n {np.array([x2[-1], y2
    [-1], z2[-1]]).reshape(-1, 1)}")
```

## Python Code (3) - Plotting points

```
fig = plt.figure()
ax = fig.add_subplot(111, projection = '3d')
ax.plot(x, y, z, label = 'Line through OR')
ax.plot(x2, y2, z2, label = 'Direction cosines of OR')

ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
ax.set_zlabel('$z$')
ax.legend(loc='best')
ax.grid()
ax.axis('equal')
```

## Python Code (4) - Labelling points

```
line_coords = np.array([[x[0], y[0], z[0]], [x[-1], y[-1], z[-1]], [x2[-1], y2[-1], z2[-1]]])

ax.scatter(line_coords[:, 0], line_coords[:, 1], line_coords[:, 2])

vert_labels = ['O', 'R', 'X']
for i, txt in enumerate(vert_labels):
    ax.text(line_coords[i][0], line_coords[i][1], line_coords[i][2], txt, color='red')
```

## Python Code (5) - 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

