MATGEO Presentation: 1.10.18

Subhodeep Chakraborty
ee25btech11055,
IIT Hyderabad.

September 6, 2025

Problem Statement

Write the direction ratios of the vector $\mathbf{a} = \hat{i} + \hat{j} - \hat{k}$ and hence calculate its direction cosines.

Direction Ratios

Given vector:

$$\mathbf{a} = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} \tag{3.1}$$

$\therefore$ The direction ratios are 1, 1 and -1.

Direction Cosines

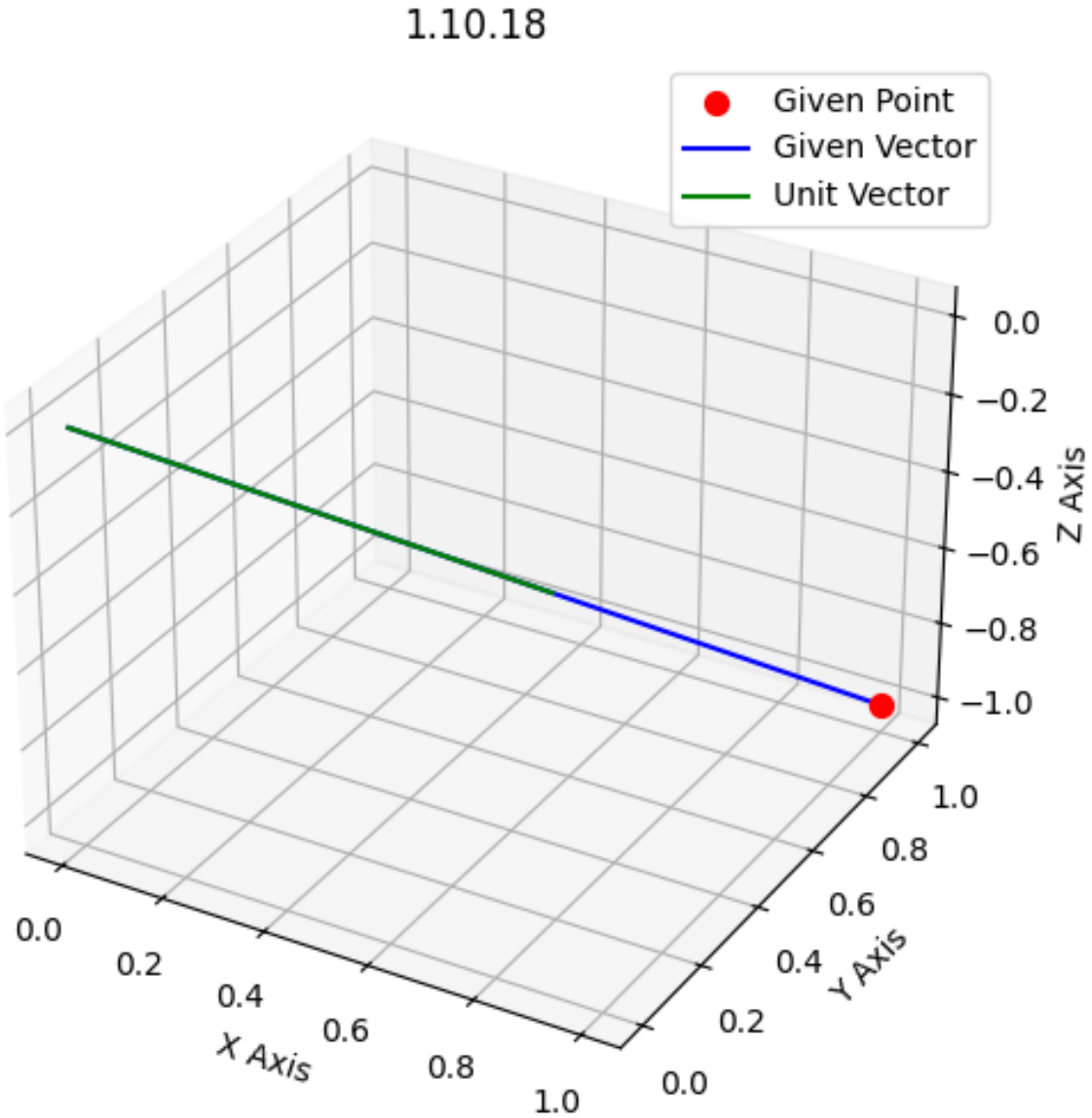
Now,

$$\|\mathbf{a}\| = \sqrt{3}$$

$$\implies \frac{\mathbf{a}}{\|\mathbf{a}\|} = \begin{pmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{-1}{\sqrt{3}} \end{pmatrix}$$

Thus we see that the direction cosines are $1/\sqrt{3}$, $1/\sqrt{3}$ and $-1/\sqrt{3}$.

Plot


1.10.18
Given Point
Given Vector
Unit Vector
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
Z Axis
0.0
0.2
0.4
0.6
0.8
1.0
X Axis
0.0
0.2
0.4
0.6
0.8
1.0
Y Axis

C code for generating points on line

```
void point_gen(const double* P1, const double* P2, double t, double*
     result_point) {
    result_point[0] = P1[0] + t * (P2[0] - P1[0]);
    result_point[1] = P1[1] + t * (P2[1] - P1[1]);
    result_point[2] = P1[2] + t * (P2[2] - P1[2]);
}
```

Python code for plotting using C

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

lib = ctypes.CDLL("./line.so")

get_point = lib.point_gen
get_point.argtypes = [
    ctypes.POINTER(ctypes.c_double), # P1
    ctypes.POINTER(ctypes.c_double), # P2
    ctypes.c_double, # t
    ctypes.POINTER(ctypes.c_double), # result_point
]
get_point.restype = None
```

Python code for plotting using C

```
DoubleArray3 = ctypes.c_double * 3
P1_arr = DoubleArray3(0, 0, 0)
P2_arr = DoubleArray3(1, 1, -1)

t_values = np.linspace(0, 1, 100)
line_points_x, line_points_y, line_points_z = [], [], []

for t in t_values:
    result_arr = DoubleArray3()

    get_point(P1_arr, P2_arr, t, result_arr)

    line_points_x.append(result_arr[0])
    line_points_y.append(result_arr[1])
    line_points_z.append(result_arr[2])

point = np.array([1, 1, -1])
```

Python code for plotting using C

```
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection="3d")
x, y, z = point
ax.scatter(
    x,
    y,
    z,
    color="red",
    s=50,
    label="Given Point",
)
```

Python code for plotting using C

```
ax.plot(
    line_points_x,
    line_points_y,
    line_points_z,
    color="blue",
    label="Given Vector",
)

unit_vec = point / LA.norm(point)
unit = list(unit_vec)
x, y, z = unit
P3_arr = DoubleArray3(x, y, z)
t_values = np.linspace(0, 1, 100)
line_points_x, line_points_y, line_points_z = [], [], []
```

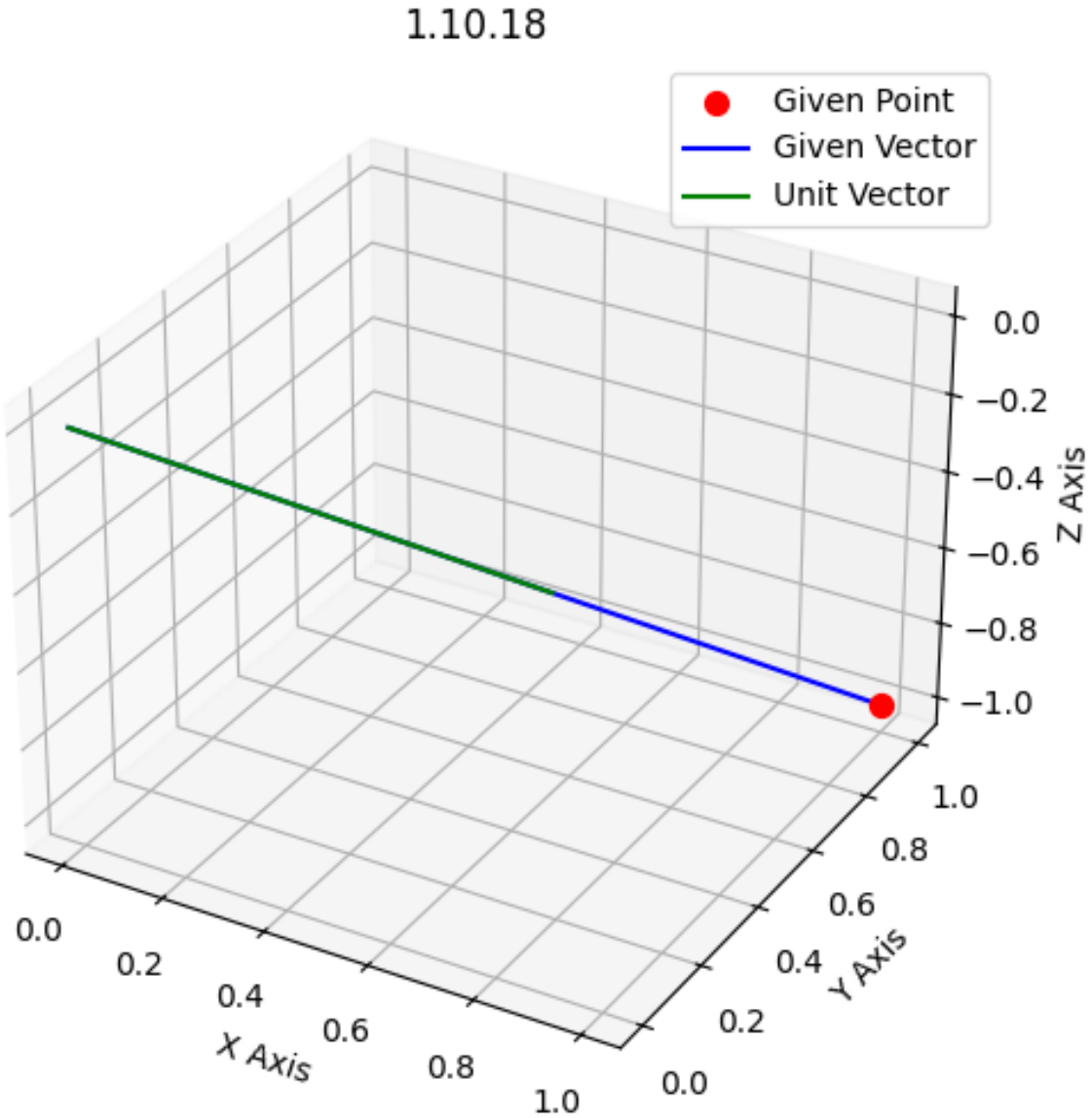
Python code for plotting using C

```
for t in t_values:
    result_arr = DoubleArray3()

    get_point(P1_arr, P3_arr, t, result_arr)

    line_points_x.append(result_arr[0])
    line_points_y.append(result_arr[1])
    line_points_z.append(result_arr[2])

ax.plot(
    line_points_x,
    line_points_y,
    line_points_z,
    color="green",
    label="Unit Vector",
)
```

Python code for plotting using C

```
   ax.set_xlabel("X Axis")
ax.set_ylabel("Y Axis")
ax.set_zlabel("Z Axis")
ax.set_title("1.10.18")
ax.legend()
ax.grid(True)

plt.savefig("../figs/plot.png")
plt.show()
```

Plot


1.10.18
Given Point
Given Vector
Unit Vector
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
Z Axis
0.0
0.2
0.4
0.6
0.8
1.0
X Axis
0.0
0.2
0.4
0.6
0.8
1.0
Y Axis

Pure Python Code for Plotting

```
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

vec = np.array([1, 1, -1]).reshape(-1, 1)

# Solving
print(f"Direction ratios are {vec[0]}, {vec[1]}, {vec[2]}")
unit = vec / LA.norm(vec)
print(f"Direction cosines are {unit[0]}, {unit[1]}, {unit[2]}")
```

Pure Python Code for Plotting

```
# Plotting

fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(111, projection="3d")

x, y, z = vec
ax.scatter(x, y, z, color="red", s=50, label="Given Point")
ax.quiver(
    0, 0, 0, x, y, z, color="blue", arrow_length_ratio=0.1, label="Position
        Vector"
)
x, y, z = unit
ax.quiver(0, 0, 0, x, y, z, color="green", arrow_length_ratio=0.1, label="
    Unit Vector")
```

Pure Python Code for Plotting

```
 ax.set_xlabel("X-axis")
ax.set_ylabel("Y-axis")
ax.set_zlabel("Z-axis")
ax.set_title("1.10.18")
ax.legend()
ax.grid(True)

plt.savefig("../figs/python.png")
plt.show()
```

Pure Python Plot