MATGEO Presentation: 2.5.18

Subhodeep Chakraborty
ee25btech11055,
IIT Hyderabad.

September 16, 2025

Problem Statement

Write the direction ratios of the vector $\mathbf{a} = \hat{\imath} + \hat{\jmath} - \hat{k}$ 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.

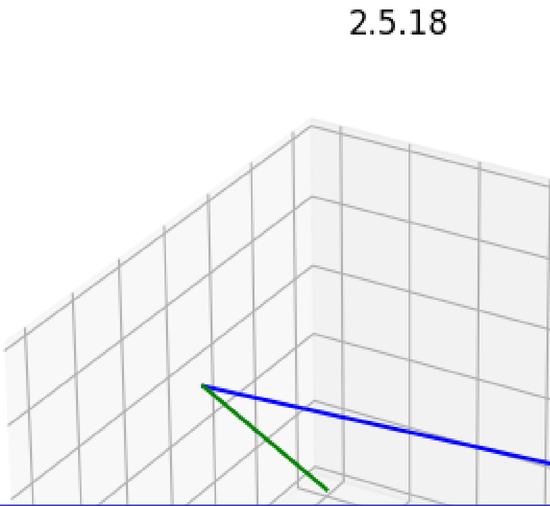
Direction Cosines

Now,

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

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

Plot



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]);
}
```

C Code for vector operations

```
  void vec_sum(const double* vec1, const double* vec2, double* sum)
      {
    sum[0] = vec1[0] + vec2[0];
    sum[1] = vec1[1] + vec2[1];
    sum[2] = vec1[2] + vec2[2];
}
void vec_diff(const double* vec1, const double* vec2, double* diff) {
    diff[0] = vec1[0] - vec2[0];
    diff[1] = vec1[1] - vec2[1];
    diff[2] = vec1[2] - vec2[2];
}
```

Python code for plotting using C

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

libline = ctypes.CDLL("./line.so")
libvec = ctypes.CDLL("./vector.so")

get_point = libline.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
```

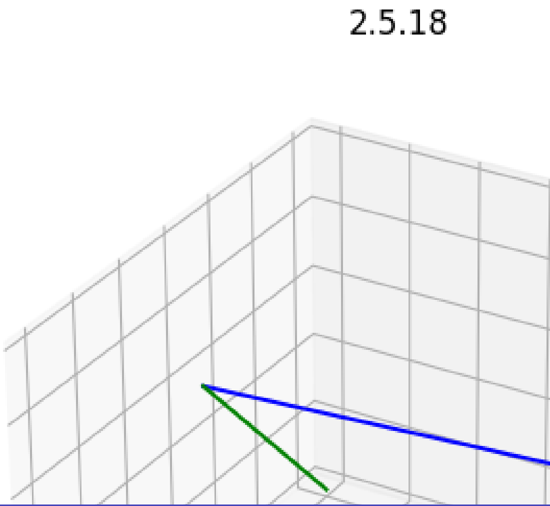
```
add = libvec.vec_sum
add.argtypes = [
    ctypes.POINTER(ctypes.c_double), # vec1
    ctypes.POINTER(ctypes.c_double), # vec2
    ctypes.POINTER(ctypes.c_double), # sum
]
add.restype = None
diff = libvec.vec_diff
diff.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double)]
diff.restype = None
DoubleArray3 = ctypes.c_double * 3
o = DoubleArray3(0, 0, 0)
a = DoubleArray3(1, 2, -3)
b = DoubleArray3(3, -1, 2)
c = DoubleArray3()
d = DoubleArray3()
```

```
add(a, b, c)
diff(a, b, d)
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection="3d")
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(o, c, 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="blue",
    label="a+b",
)
```

```
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(o, d, 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="a-b",
)
```

```
ax.set_xlabel("X Axis")
ax.set_ylabel("Y Axis")
ax.set_zlabel("Z Axis")
ax.set_title("2.5.18")
ax.legend()
ax.grid(True)

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

Plot



Pure Python code

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

a = np.array([1, 2, -3]).T
b = np.array([3, -1, 2]).T

# Solving
c = a + b
d = a - b
result = (c.T) @ d
if result == 0:
    print("a+b and a-b are perpendicular")
else:
    print("a+b and a-b are not perpendicular")
```

Pure Python code

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

ax.quiver(0, 0, 0, c[0], c[1], c[2], color="red", label="a+b")
ax.quiver(0, 0, 0, d[0], d[1], d[2], color="blue", label="a−b")

ax.set_xlabel("X−axis")
ax.set_ylabel("Y−axis")
ax.set_zlabel("Z−axis")
ax.set_title("2.5.18")
ax.set_xlim([−5, 5])
ax.set_ylim([−5, 5])
ax.set_zlim([−5, 5])
ax.legend()
ax.grid(True)
plt.savefig("../figs/python.png")
```

Plot