Problem 4.11.1

ee25btech11023-Venkata Sai

September 23, 2025

Problem

Find the coordinates of the point where the line $\frac{x-1}{3} = \frac{y+4}{7} = \frac{z+4}{2}$ cuts the XY-plane

Declaration of variables

The line equation is

$$\mathbf{r} = \mathbf{a} + t\mathbf{b} \tag{1.1}$$

where $\mathbf{a}$ is the point on line and $\mathbf{b}$ is the direction vector

$$\mathbf{a} = \begin{pmatrix} 1 \\ -4 \\ -4 \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} 3 \\ 7 \\ 2 \end{pmatrix} \tag{1.2}$$

The normal vector to XY plane is

$$\mathbf{n} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{1.3}$$

The plane equation of the XY-plane is

$$\mathbf{n}^T\mathbf{x} = 0 \implies \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}\mathbf{x} = 0 \tag{1.4}$$

Solving

Substituting the line into the plane equation gives

$$\mathbf{n}^T(\mathbf{a} + t\mathbf{b}) = 0 \tag{1.5}$$

$$\mathbf{n}^T\mathbf{a} + t\left(\mathbf{n}^T\mathbf{b}\right) = 0 \tag{1.6}$$

$$t\left(\mathbf{n}^T\mathbf{b}\right) = -\mathbf{n}^T\mathbf{a} \tag{1.7}$$

$$t = -\frac{\mathbf{n}^T\mathbf{a}}{\mathbf{n}^T\mathbf{b}} \tag{1.8}$$

$$t = -\frac{\begin{pmatrix}0 & 0 & 1\end{pmatrix}\begin{pmatrix}1\\-4\\-4\end{pmatrix}}{\begin{pmatrix}0 & 0 & 1\end{pmatrix}\begin{pmatrix}3\\7\\2\end{pmatrix}} = -\frac{-4}{2} \tag{1.9}$$

$$\implies t = 2 \tag{1.10}$$

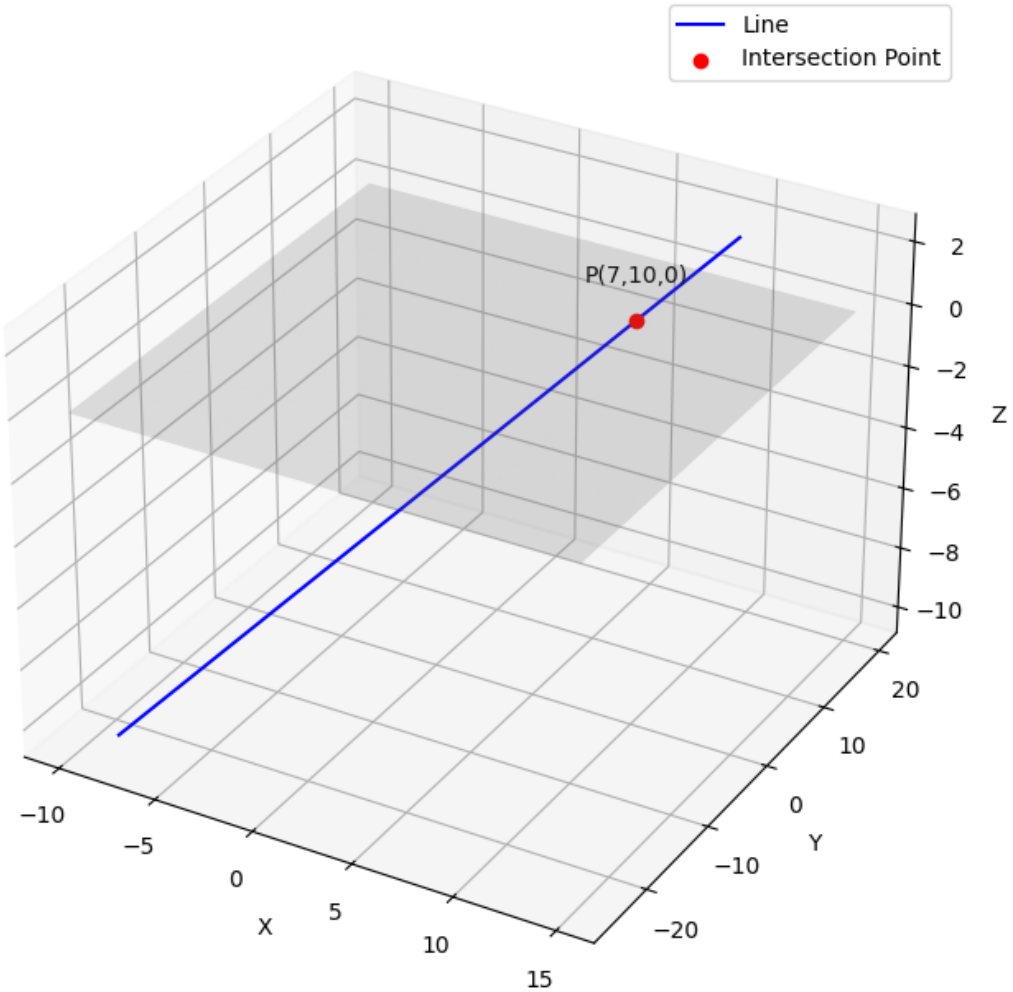
Conclusion

The intersection point is

$$\mathbf{r} = \mathbf{a} + t\mathbf{b} = \begin{pmatrix} 1 \\ -4 \\ -4 \end{pmatrix} + 2 \begin{pmatrix} 3 \\ 7 \\ 2 \end{pmatrix} \tag{1.11}$$

$$\mathbf{r} = \begin{pmatrix} 7 \\ 10 \\ 0 \end{pmatrix} \tag{1.12}$$

Plot


Line
Intersection Point
P(7,10,0)
2
0
−2
−4
−6
−8
−10
Z
20
10
0
−10
−20
Y
−10
−5
0
5
10
15
X

C Code

```
void get_intersection_data(double* out_data) {
    double point_a[3] = {1.0, -4.0, -4.0};
    double dir_d[3] = {3.0, 7.0, 2.0};
    double lambda = 2.0;

    double Px = point_a[0] + lambda * dir_d[0];
    double Py = point_a[1] + lambda * dir_d[1];
    double Pz = point_a[2] + lambda * dir_d[2];

    out_data[0] = Px;
    out_data[1] = Py;
    out_data[2] = Pz;
    out_data[3] = point_a[0];
    out_data[4] = point_a[1];
    out_data[5] = point_a[2];
    out_data[6] = dir_d[0];
    out_data[7] = dir_d[1];
    out_data[8] = dir_d[2];
}
```

Python Code for Calling

```
import ctypes
import numpy as np

def get_line_data_from_c():

    lib = ctypes.CDLL('./coord.so')
    double_array_9 = ctypes.c_double * 9
    lib.get_intersection_data.argtypes = [ctypes.POINTER(ctypes.
        c_double)]

    out_data_c = double_array_9()
    lib.get_intersection_data(out_data_c)
    all_data = np.array(out_data_c)
    # Split the data into the three vectors
    intersection_P = all_data[0:3]
    point_a = all_data[3:6]
    dir_d = all_data[6:9]

    return intersection_P, point_a, dir_d
```

Python Code for Plotting

```
#Code by GVV Sharma
#September 12, 2023
#Revised July 21, 2024
#released under GNU GPL
import sys #for path to external scripts
sys.path.insert(0, '/workspaces/urban-potato/matgeo/codes/
    CoordGeo/')
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

#local imports
from line.funcs import *
from triangle.funcs import *

from call import get_line_data_from_c
```

Python Code for Plotting

```
P, point_a, dir_d = get_line_data_from_c()
 line_points=point_a+np.array([-3,3]).reshape(-1,1)*dir_d
fig=plt.figure(figsize=(8,8))
ax=fig.add_subplot(111,projection='3d')
X_plane=np.linspace(-10,15,5)
Y_plane=np.linspace(-25,20,10)
X_plane,Y_plane = np.meshgrid(X_plane,Y_plane)
Z_plane=np.zeros_like(X_plane)
ax.plot_surface(X_plane,Y_plane,Z_plane,alpha=0.2,color='gray')
ax.plot(line_points[:,0],line_points[:,1],line_points[:,2],color=
    'b',label='Line')
```

Python Code for Plotting

```
ax.scatter(P[0], P[1], P[2], color='r', s=35, label='Intersection
    Point')
ax.text(P[0], P[1], P[2] + 1.2, f'P({P[0]:.0f}, {P[1]:.0f}, {P
    [2]:.0f})', ha='center')

ax.set_title('Intersection of a Line with the XY Plane')
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.legend()
plt.show()
plt.savefig('../figs/fig1.png')
```