Matgeo Presentation - Problem 4.11.33

ee25btech11056 - Suraj.N

September 9, 2025

Problem Statement

Find the length of the intercept cut off by the plane $2x + y - z = 5$ on the X axis.

Data

Description	**Value**
Plane	$2x + y - z = 5 \iff \begin{pmatrix} 2 & 1 & -1 \end{pmatrix} \mathbf{x} = 5$

Table : Plane

Solution

The equation of plane is

$$\mathbf{n}^\top \mathbf{x} = c \tag{0.1}$$

$$\begin{pmatrix} 2 & 1 & -1 \end{pmatrix} \mathbf{x} = 5 \tag{0.2}$$

The X-intercept of the plane is of the form

$$\mathbf{P} = \begin{pmatrix} a \\ 0 \\ 0 \end{pmatrix}. \tag{0.3}$$

P lies on the plane,

$$\begin{pmatrix} 2 & 1 & -1 \end{pmatrix} \mathbf{P} = 5 \tag{0.4}$$

$$\begin{pmatrix} 2 & 1 & -1 \end{pmatrix} \begin{pmatrix} a \\ 0 \\ 0 \end{pmatrix} = 5 \tag{0.5}$$

$$2a + 0 + 0 = 5 \tag{0.6}$$

$$a = \frac{5}{2} \tag{0.7}$$

Thus, the intercept point is

$$\mathbf{P} = \begin{pmatrix} \frac{5}{2} \\ 0 \\ 0 \end{pmatrix} \tag{0.8}$$

Answer: The intercept length is $\frac{5}{2}$.

Plot

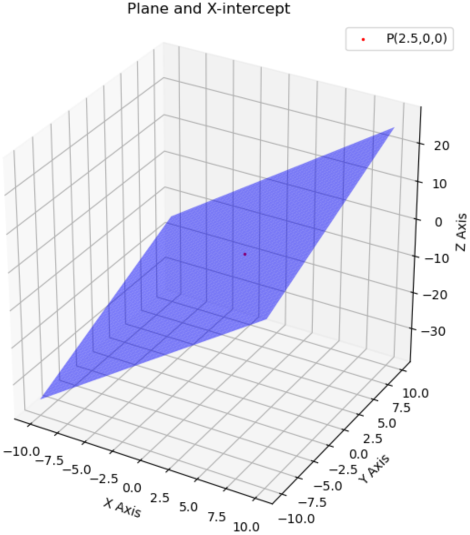

Fig : Plane

C Code: points.c

```
#include <stdio.h>

double intercept(double n[], double c) {

  double a;

  a = (c / n[0]);

  return a;
}
```

Python: call_c.py

```
import ctypes
import sys
import os
import numpy as np
import matplotlib.pyplot as plt

#to save the figure in figs folder
figs_folder = os.path.join("..","figs")

#to load the shared object
lib = ctypes.CDLL("./points.so")

#defining the function signatures
lib.intercept.restype = ctypes.c_double
lib.intercept.argtypes = [ctypes.POINTER(ctypes.c_double),ctypes.c_double]

#normal vector and c
n = (ctypes.c_double*3)(2,1,-1)
c = 5.0

#calling the c function
x_intercept = lib.intercept(n,c)

#writing the x_intercept as a numpy array
point = np.array([x_intercept,0,0])

#creating a meshgrid for the plane
x = np.linspace(-10,10,100)
y = np.linspace(-10,10,100)

x,y = np.meshgrid(x,y)
```

Python: call_c.py

```
#plane equation
z = (c - n[0]*x-n[1]*y)/n[2]

#plotting
fig=plt.figure(figsize=(8,6))
ax = fig.add_subplot(111,projection="3d")
ax.set_box_aspect([1,1,1]) # Fix aspect ratio : to enforce equal scaling across axes so that xintercept lies
        on x axis.

#plane
ax.plot_surface(x,y,z,alpha=0.5,color='blue',edgecolor='none')

#intercept point
ax.scatter(*point,color='red',s=2,label=f"P({x_intercept},0,0)")

#axes labels
ax.set_xlabel("X␣Axis")
ax.set_ylabel("Y␣Axis")
ax.set_zlabel("Z␣Axis")
ax.set_title("Plane␣and␣X-intercept")

ax.legend()
ax.grid(True)

plt.tight_layout()
fig.savefig(os.path.join(figs_folder,"intercept.png"))
plt.show()
```

Python: plot.py

```
import os
import numpy as np
import matplotlib.pyplot as plt

#to save the figure in figs folder
figs_folder = os.path.join("..","figs")

#normal vector and c
n = np.array([2,1,-1])
c = 5.0

#x intercept
x_intercept = c/n[0]

#writing the x_intercept as a numpy array
point = np.array([x_intercept,0,0])

#creating a meshgrid for the plane
x = np.linspace(-10,10,100)
y = np.linspace(-10,10,100)

x,y = np.meshgrid(x,y)

#plane equation
z = (c - n[0]*x - n[1]*y)/n[2]

#plotting
fig=plt.figure(figsize=(8,6))
ax = fig.add_subplot(111,projection="3d")
ax.set_box_aspect([1,1,1]) # Fix aspect ratio : to enforce equal scaling across axes so that xintercept lies
        on x axis.
```

Python: plot.py

```
#plane
ax.plot_surface(x,y,z,alpha=0.5,color='blue',edgecolor='none')

#intercept point
ax.scatter(*point,color='red',s=2,label=f"P({x_intercept},0,0)")

#axes labels
ax.set_xlabel("X␣Axis")
ax.set_ylabel("Y␣Axis")
ax.set_zlabel("Z␣Axis")
ax.set_title("Plane␣and␣X-intercept")

ax.legend()
ax.grid(True)

plt.tight_layout()
fig.savefig(os.path.join(figs_folder,"intercept.png"))
plt.show()
```