TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING

HIMALAYA COLLEGE OF ENGINEERING
CHYASAL, LALITPUR

Lab Report No: -01

Title: - Basic Plotting Using Matplotlib (Python)

Submitted by: -

Name: - Diwas Pokhrel

Roll no: - HCE081BEI014

Date of submission: - 2082/08/26

OBJECTIVE

1. To learn basic Python syntax and about Matplotlib in Python.
2. To understand and implement the DDA algorithm fir drawing a straight line.
3. To practice plotting different shapes using Matplotlib for visualization.

SOFTWARE REQUIREMENTS

- Python 3.14
- Vs Code
- Python libraries: matplotlib

THEORY

: Line Drawing Using the DDA Algorithm

In computer graphics, drawing a straight line between two points is a fundamental task. The Digital Differential Analyzer (DDA) algorithm provides a simple method to compute all the intermediate points required to plot a line on a raster display. It works by calculating small step-by-step increments between the starting and ending coordinates.
For two points (x1, y1) and (x2, y2), the algorithm begins by finding the differences:

$$dx = x2 - x1$$

$$dy = y2 - y1$$

The total number of steps is chosen as the larger value between |dx| and |dy|, ensuring smooth plotting regardless of the line's slope. The increments are then calculated as:

$$xinc = dx / steps$$

$$yinc = dy / steps.$$

Starting from the initial coordinate, these increments are repeatedly added to generate every point on the line. The values are rounded to obtain pixel positions, which are then plotted. This approach demonstrates the use of arithmetic operations, incremental computation, and plotting through Python lists and Matplotlib.

Rectangle Drawing Using Coordinate Geometry

A rectangle on a 2D plane can be generated using the coordinates of any two opposite vertices. When the diagonal points are known, the other two corners are obtained by combining the x- and y-values of the given points.
For points (x1, y1) and (x2, y2), the remaining corners become (x2, y1)and (x1, y2).
To plot the rectangle, the points are arranged in proper sequence, and the first point is repeated at the end to close the shape. These coordinate lists are then plotted using Matplotlib with markers and

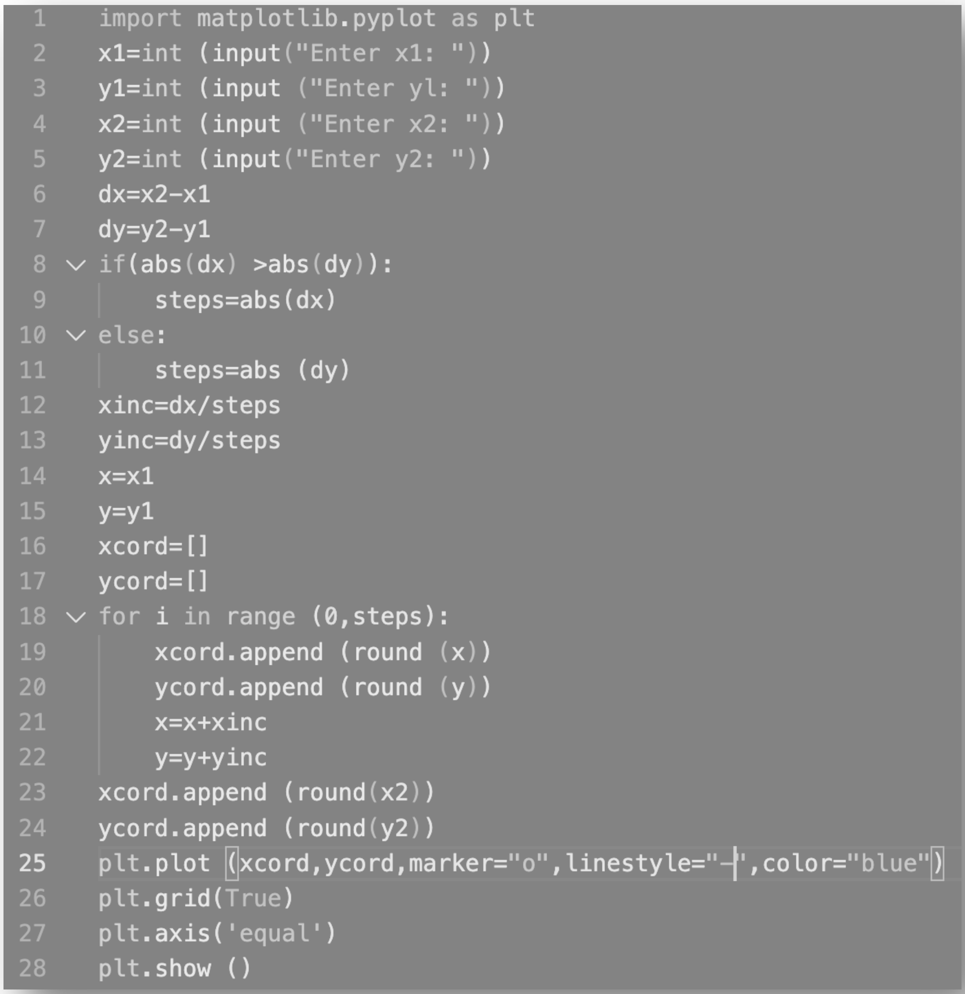
dashed lines. This method shows how coordinate geometry can be used to construct shapes and how Python can visualize them using functions such as plot(), grid(), and axis('equal').

Importance of Plotting in Computer Graphics

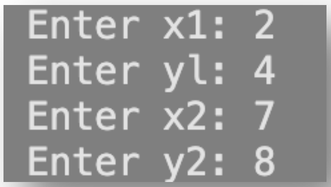
Plotting is an essential part of computer graphics because it helps in visualizing geometric structures and understanding how algorithms work. Using plotting libraries like Matplotlib, students can see the effect of their calculations directly on a graph. It also helps visualize relationships between coordinates, slopes, and shapes, making mathematical concepts easier to understand.
Plotting in 2D serves as the foundation for advanced graphical topics such as transformations, clipping, windowing, and even 3D rendering. By learning simple plotting methods early, students build skills that support future computer graphics applications.

LAB ASSIGNMENTS

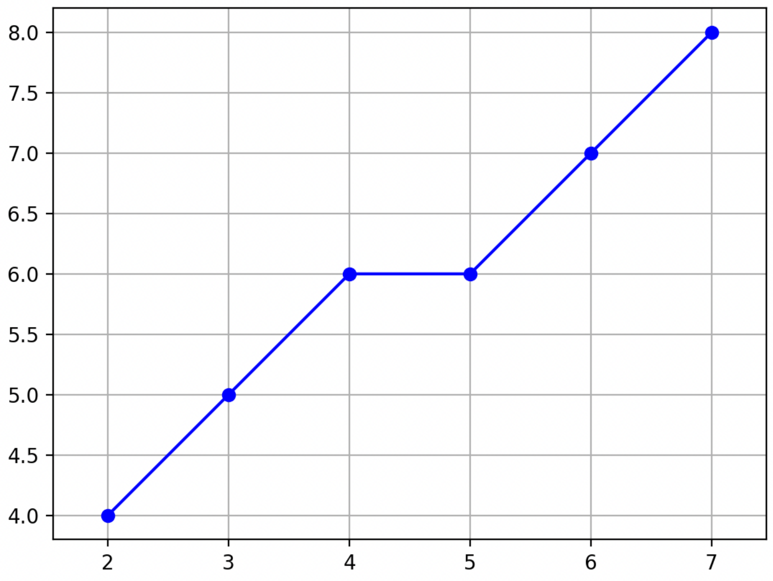
1) Write a python program to draw a straight line between two points using the DDA algorithm.



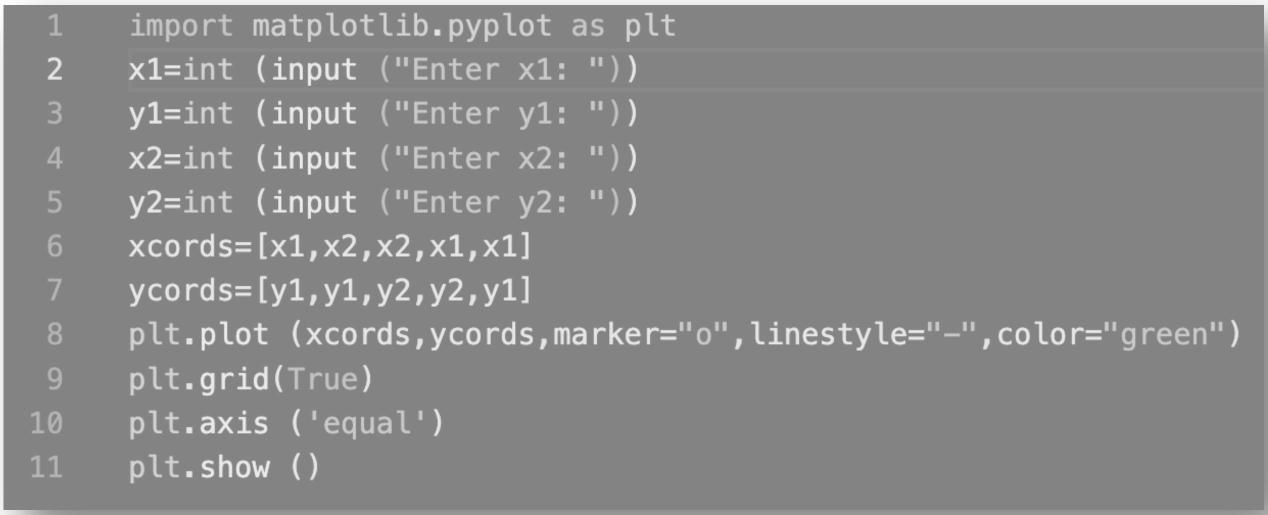
```python
import matplotlib.pyplot as plt
x1=int (input("Enter x1: "))
y1=int (input ("Enter yl: "))
x2=int (input ("Enter x2: "))
y2=int (input("Enter y2: "))
dx=x2-x1
dy=y2-y1
if(abs(dx) >abs(dy)):
    steps=abs(dx)
else:
    steps=abs (dy)
xinc=dx/steps
yinc=dy/steps
x=x1
y=y1
xcord=[]
ycord=[]
for i in range (0,steps):
    xcord.append (round (x))
    ycord.append (round (y))
    x=x+xinc
    y=y+yinc
xcord.append (round(x2))
ycord.append (round(y2))
plt.plot (xcord,ycord,marker="o",linestyle="-",color="blue")
plt.grid(True)
plt.axis('equal')
plt.show ()
```



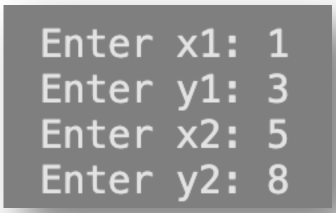
```
Enter x1: 2
Enter yl: 4
Enter x2: 7
Enter y2: 8
```



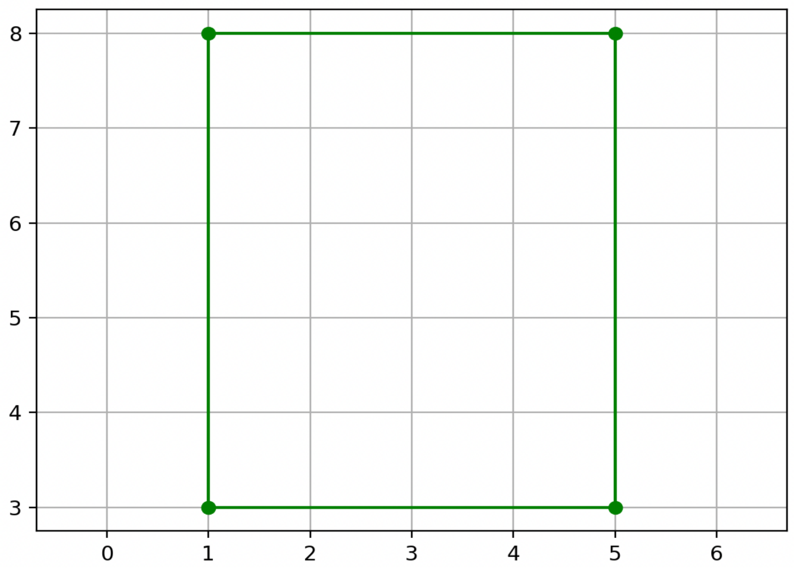
2) Write a python program to draw a rectangle on a 2D plane using the coordinates of two diagonally opposite corners.



```python
import matplotlib.pyplot as plt
x1=int (input ("Enter x1: "))
y1=int (input ("Enter y1: "))
x2=int (input ("Enter x2: "))
y2=int (input ("Enter y2: "))
xcords=[x1,x2,x2,x1,x1]
ycords=[y1,y1,y2,y2,y1]
plt.plot (xcords,ycords,marker="o",linestyle="-",color="green")
plt.grid(True)
plt.axis ('equal')
plt.show ()
```



```
Enter x1: 1
Enter y1: 3
Enter x2: 5
Enter y2: 8
```



Discussion

In this lab, two basic computer graphics tasks—line drawing and rectangle construction—were performed using Python and Matplotlib. The DDA line drawing algorithm helped visualize how incremental calculations generate a continuous line between two coordinate points. By observing the plotted output, it became clear how rounding decisions and step size affect the smoothness and accuracy of the line.The rectangle plotting task demonstrated the practical use of coordinate geometry in constructing shapes. By using only two diagonal points, the remaining corners were derived logically, showing how geometric relationships simplify shape generation. The plotting process also illustrated how point ordering determines the final visual structure of the rectangle.

Conclusion

This lab successfully demonstrated the implementation of fundamental graphic algorithms using Python. The DDA algorithm enabled smooth and step-wise line drawing, while coordinate geometry was effectively used to construct and visualize a rectangle. Through practical coding and plotting, the concepts of incremental computation, coordinate handling, and shape visualization became clearer. By completing the lab, we gained hands-on experience with graphic plotting tools and learned how mathematical operations form the basis of computer-generated visuals. The techniques explored here provide a strong foundation for more advanced topics in computer graphics, such as transformations, clipping, and rendering.