Lab No 2

IMPLEMENTATION OF DDA LINE DRAWING ALGORITHM (PYTHON)

Computer Graphics

Instructor: Sushant Pandey

Report Submitted on: ***2082/08/28***

Department of Electronics & Computer Engineering

Himalaya College of Engineering, TU

Chyasal, Lalitpur

Objective(s):

i) To understand the Digital Differential Analyzer (DDA) line drawing algorithm.

ii) To implement the DDA algorithm in Python and visualize the generated pixels.

iii) To compare DDA with the analytical line equation.

Software(s) Required

Python 3, matplotlib, any IDE or Jupyter Notebook.

Theory

In raster graphics, a straight line must be approximated using discrete pixels. The DDA algorithm is an incremental method that uses the line slope to step along the dominant axis and compute intermediate points.

Given two endpoints (x1, y1) and (x2, y2), we define:

$$d_x = x2 - x1,\ d_y = y2 - y1$$

The number of steps is chosen as:

$$steps = \max(|dx|, |dy|)$$

The increment in each step is:

$$xinc = dx/steps$$

$$yinc = dy/steps$$

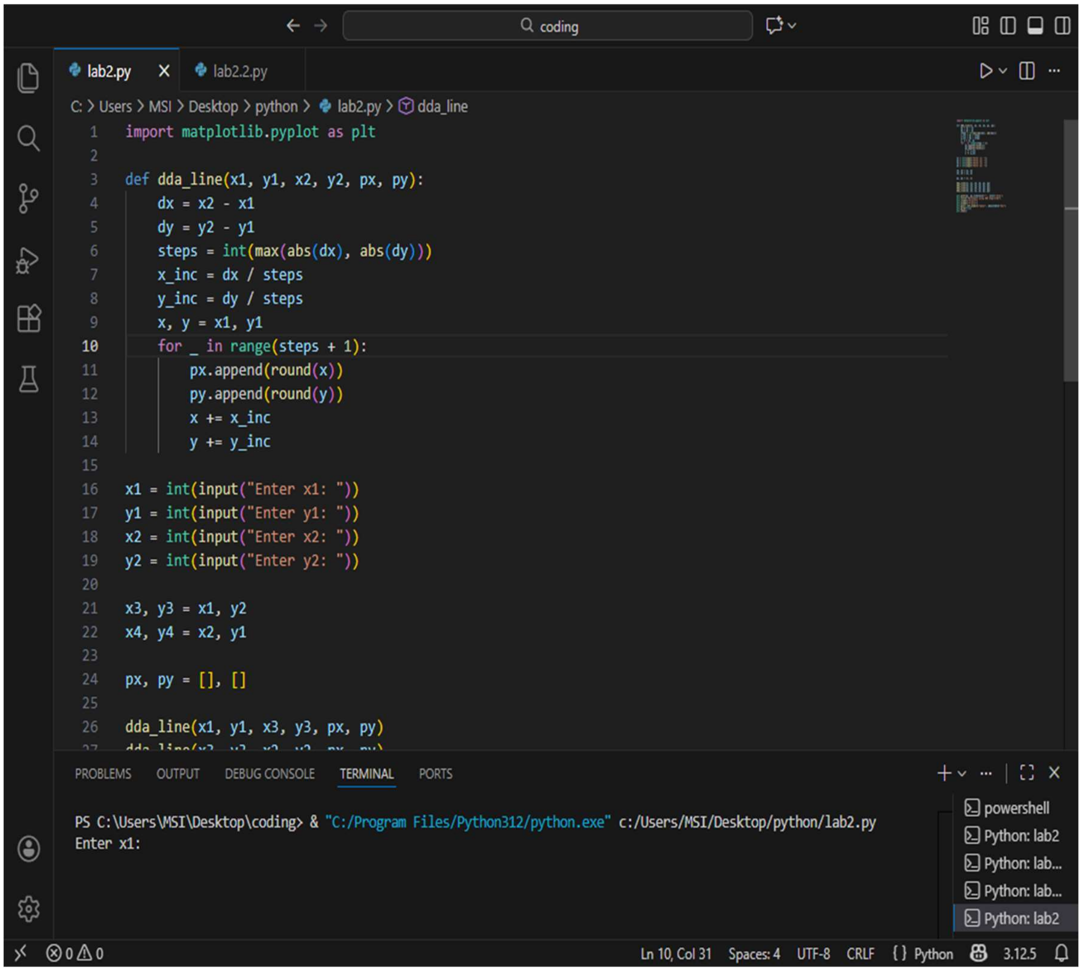
Starting from (x1, y1), the algorithm adds these increments repeatedly and rounds to the nearest pixel.

Algorithm: DDA Line Drawing

```
1. Read starting point (x1, y1) and ending point (x2, y2).
2. Compute dx, dy and number of steps.
3. Compute xinc and yinc.
4. Initialise x = x1, y = y1.
5. For k = 0 to steps:
     • Plot pixel at (round(x), round(y)).
     • Update x = x + xinc, y = y + yinc.
```

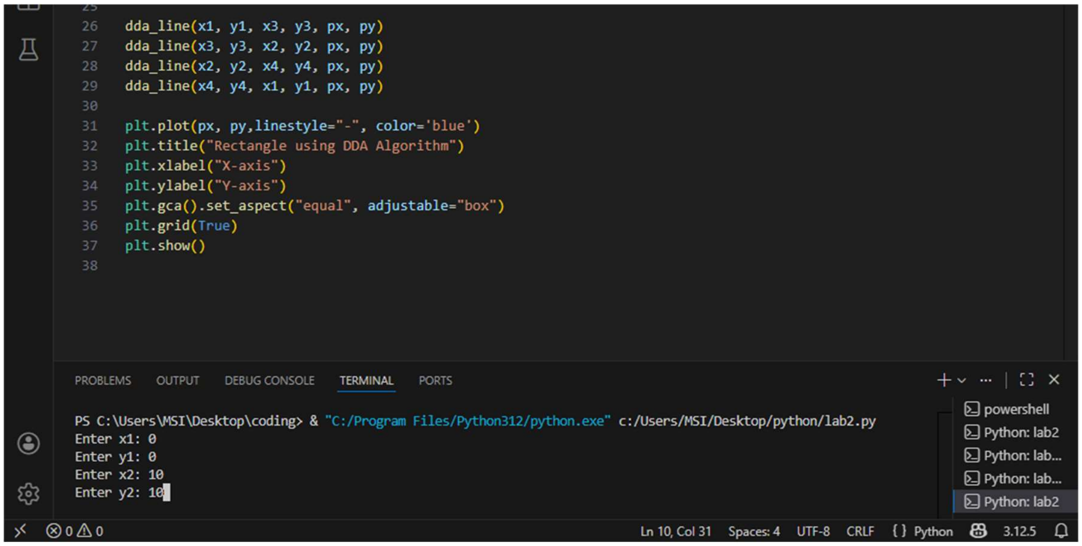
Lab Assignment

1. Extend the DDA program to draw a rectangle given two opposite corners.



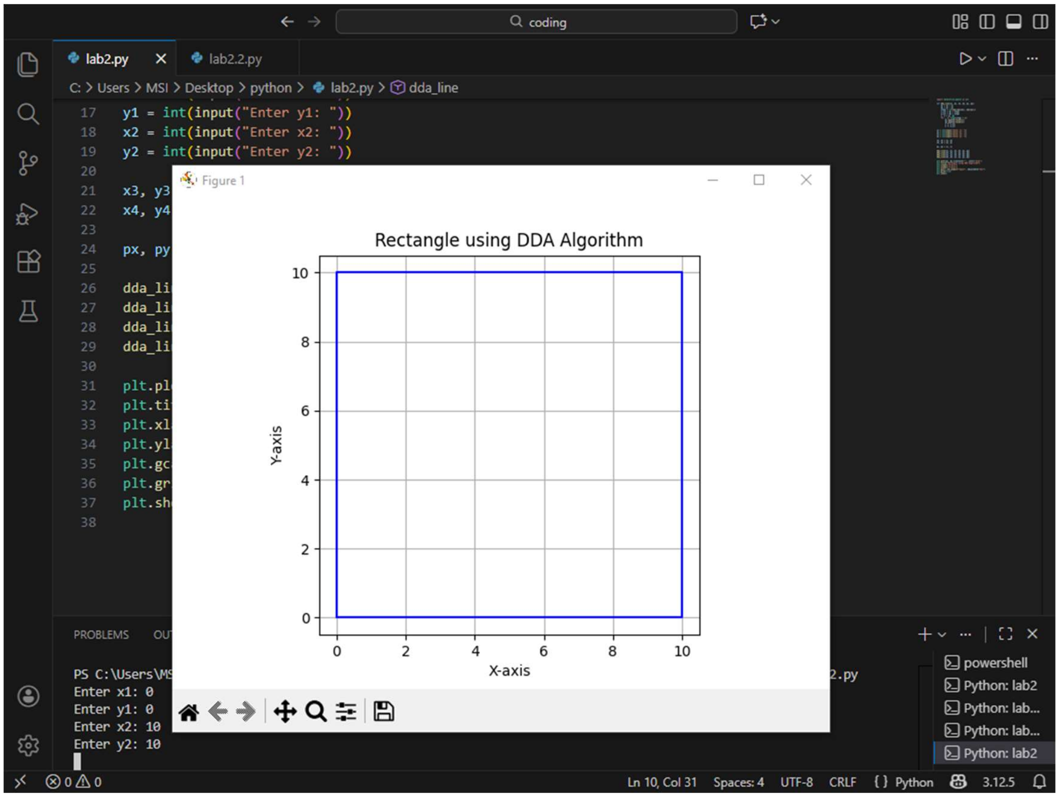
```python
import matplotlib.pyplot as plt

def dda_line(x1, y1, x2, y2, px, py):
    dx = x2 - x1
    dy = y2 - y1
    steps = int(max(abs(dx), abs(dy)))
    x_inc = dx / steps
    y_inc = dy / steps
    x, y = x1, y1
    for _ in range(steps + 1):
        px.append(round(x))
        py.append(round(y))
        x += x_inc
        y += y_inc

x1 = int(input("Enter x1: "))
y1 = int(input("Enter y1: "))
x2 = int(input("Enter x2: "))
y2 = int(input("Enter y2: "))

x3, y3 = x1, y2
x4, y4 = x2, y1

px, py = [], []

dda_line(x1, y1, x3, y3, px, py)
```

```
PS C:\Users\MSI\Desktop\coding> & "C:/Program Files/Python312/python.exe" c:/Users/MSI/Desktop/python/lab2.py
Enter x1:
```



```python
dda_line(x1, y1, x3, y3, px, py)
dda_line(x3, y3, x2, y2, px, py)
dda_line(x2, y2, x4, y4, px, py)
dda_line(x4, y4, x1, y1, px, py)

plt.plot(px, py,linestyle="-", color='blue')
plt.title("Rectangle using DDA Algorithm")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.gca().set_aspect("equal", adjustable="box")
plt.grid(True)
plt.show()
```

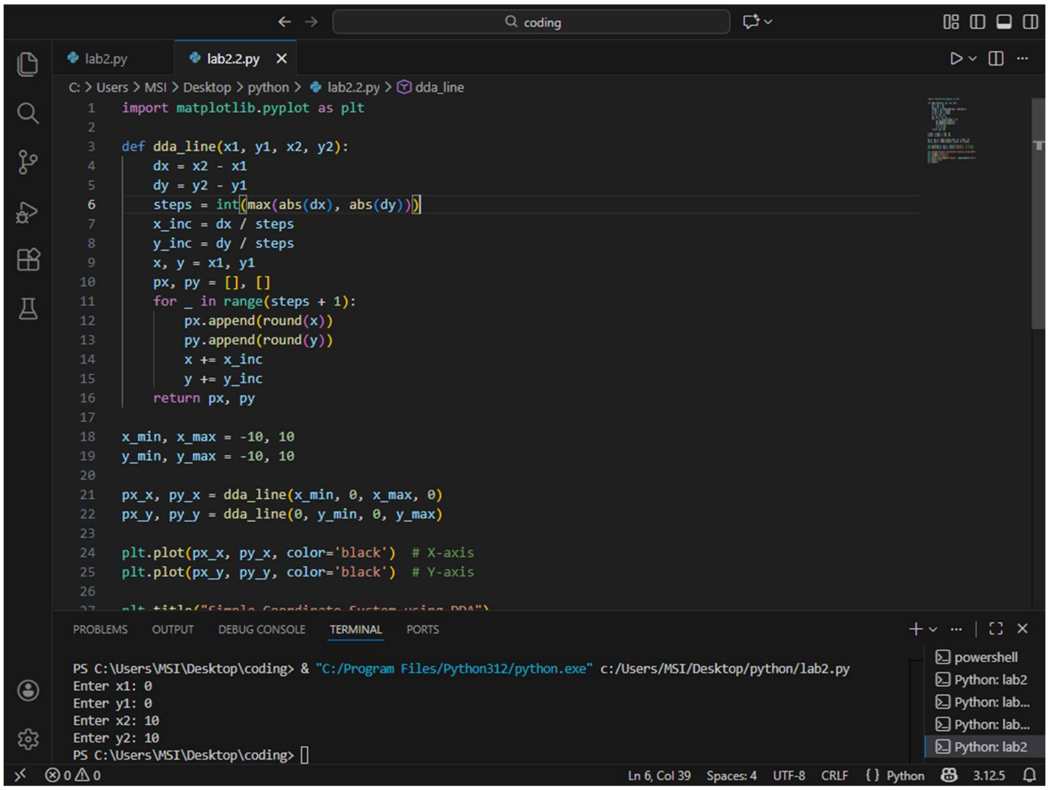
```
PS C:\Users\MSI\Desktop\coding> & "C:/Program Files/Python312/python.exe" c:/Users/MSI/Desktop/python/lab2.py
Enter x1: 0
Enter y1: 0
Enter x2: 10
Enter y2: 10
```



```python
y1 = int(input("Enter y1: "))
x2 = int(input("Enter x2: "))
y2 = int(input("Enter y2: "))

x3, y3
x4, y4

px, py

dda_li
dda_li
dda_li
dda_li

plt.pl
plt.ti
plt.xl
plt.yl
plt.gc
plt.gr
plt.sh
```

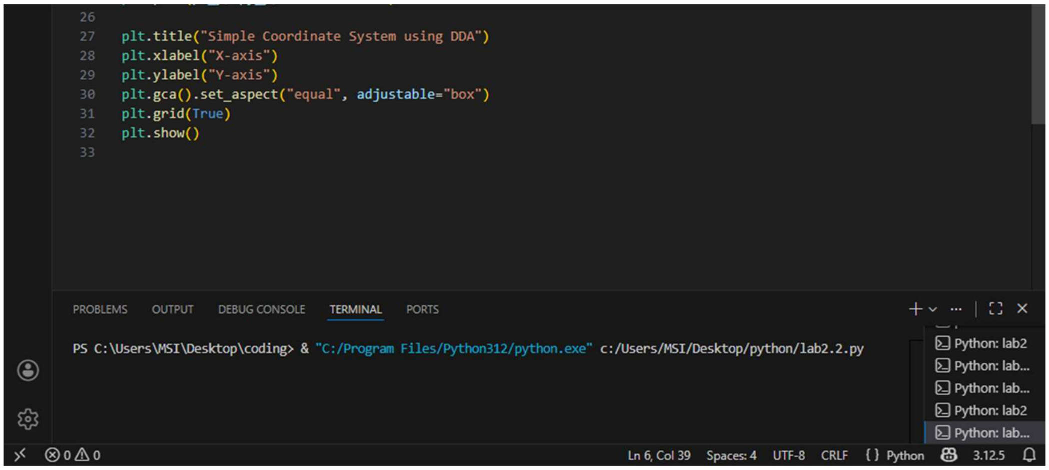
```
PS C:\Users\MS                                   2.py
Enter x1: 0
Enter y1: 0
Enter x2: 10
Enter y2: 10
```

2. Use DDA to draw the axes of a simple coordinate system (X and Y axes).



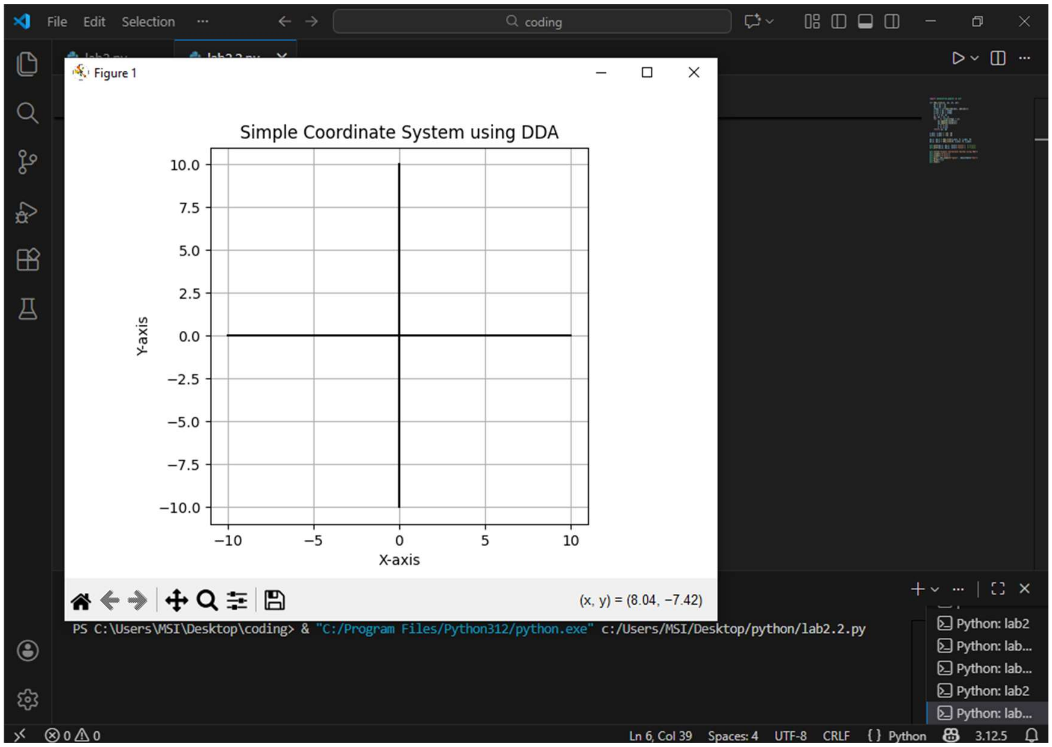
```python
import matplotlib.pyplot as plt

def dda_line(x1, y1, x2, y2):
    dx = x2 - x1
    dy = y2 - y1
    steps = int(max(abs(dx), abs(dy)))
    x_inc = dx / steps
    y_inc = dy / steps
    x, y = x1, y1
    px, py = [], []
    for _ in range(steps + 1):
        px.append(round(x))
        py.append(round(y))
        x += x_inc
        y += y_inc
    return px, py

x_min, x_max = -10, 10
y_min, y_max = -10, 10

px_x, py_x = dda_line(x_min, 0, x_max, 0)
px_y, py_y = dda_line(0, y_min, 0, y_max)

plt.plot(px_x, py_x, color='black')  # X-axis
plt.plot(px_y, py_y, color='black')  # Y-axis

plt.title("Simple Coordinate System using DDA")
```

```
PS C:\Users\MSI\Desktop\coding> & "C:/Program Files/Python312/python.exe" c:/Users/MSI/Desktop/python/lab2.py
Enter x1: 0
Enter y1: 0
Enter x2: 10
Enter y2: 10
PS C:\Users\MSI\Desktop\coding> 
```



```python

plt.title("Simple Coordinate System using DDA")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.gca().set_aspect("equal", adjustable="box")
plt.grid(True)
plt.show()

```

```
PS C:\Users\MSI\Desktop\coding> & "C:/Program Files/Python312/python.exe" c:/Users/MSI/Desktop/python/lab2.2.py
```



```
PS C:\Users\MSI\Desktop\coding> & "C:/Program Files/Python312/python.exe" c:/Users/MSI/Desktop/python/lab2.2.py
```

Conclusion

In this lab we gained the practical experience in handling geometric computations using matplotlib. We learned the concept of DDA line drawing algorithm and made the rectangle using it.