Lab 1

Name : Abdelaziz Mohamed Abdelaziz

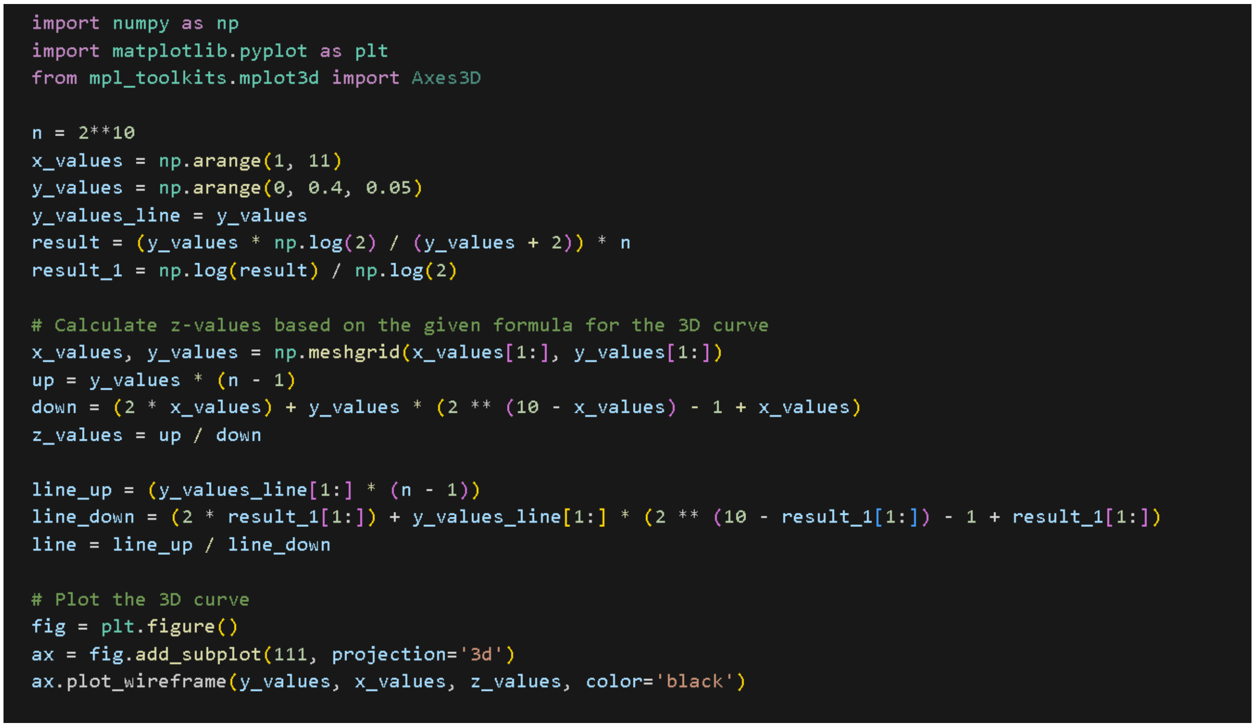
I.D : 19015941

Plotting 3D Curve :

According to values of q and y we calculate F(y,q) according to this function :

F(y,q) = y * (n-1) / 2q + y (2^(n-q) – 1 + q) and then calculate z_values and draw 3d Curve using plot_wireframe .

make meshgrid between q and y to create a 2d grid .

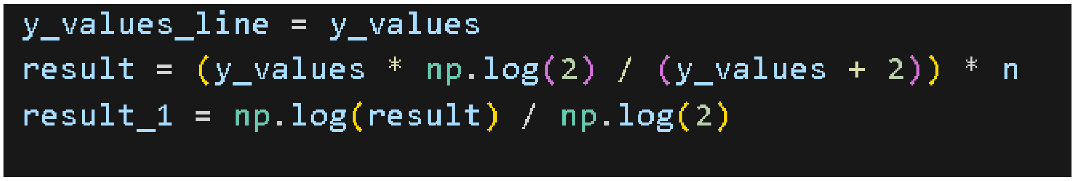


```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

n = 2**10
x_values = np.arange(1, 11)
y_values = np.arange(0, 0.4, 0.05)
y_values_line = y_values
result = (y_values * np.log(2) / (y_values + 2)) * n
result_1 = np.log(result) / np.log(2)

# Calculate z-values based on the given formula for the 3D curve
x_values, y_values = np.meshgrid(x_values[1:], y_values[1:])
up = y_values * (n - 1)
down = (2 * x_values) + y_values * (2 ** (10 - x_values) - 1 + x_values)
z_values = up / down

line_up = (y_values_line[1:] * (n - 1))
line_down = (2 * result_1[1:]) + y_values_line[1:] * (2 ** (10 - result_1[1:]) - 1 + result_1[1:])
line = line_up / line_down

# Plot the 3D curve
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_wireframe(y_values, x_values, z_values, color='black')
```

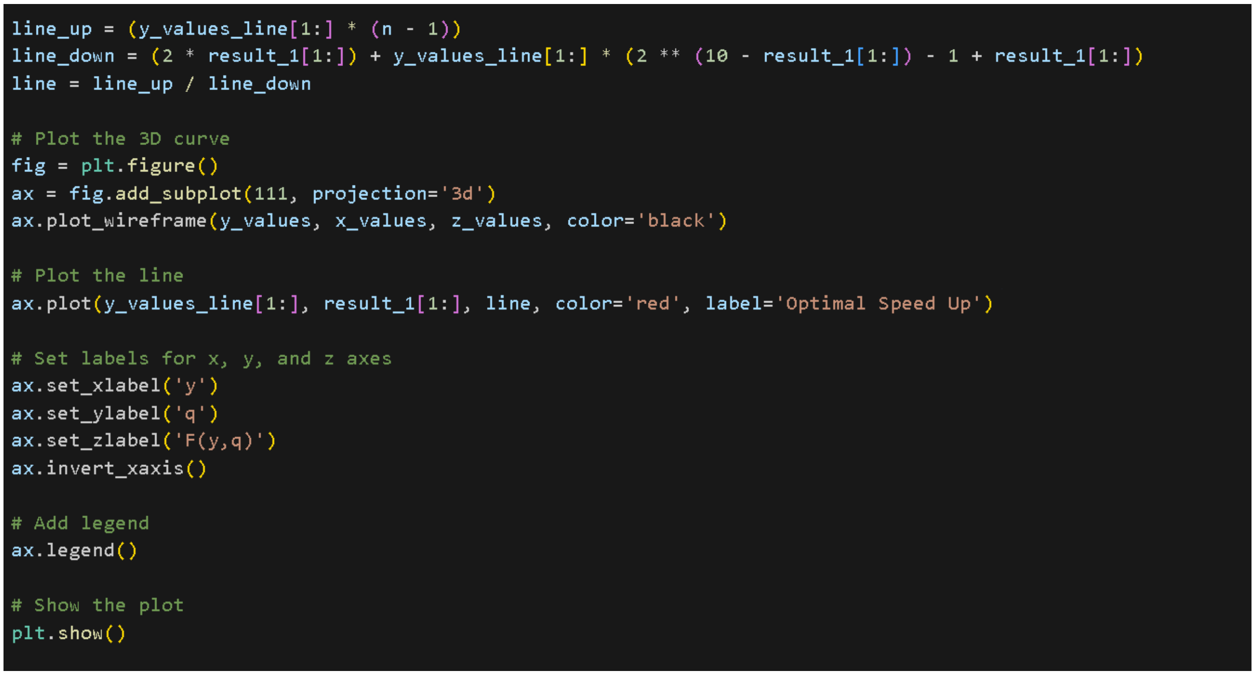
Plotting Line (optimal solution) :

We use same curve equation but we calculate f(p(y),n) in which p(y) is calculated using (y ln2/y+2) n.

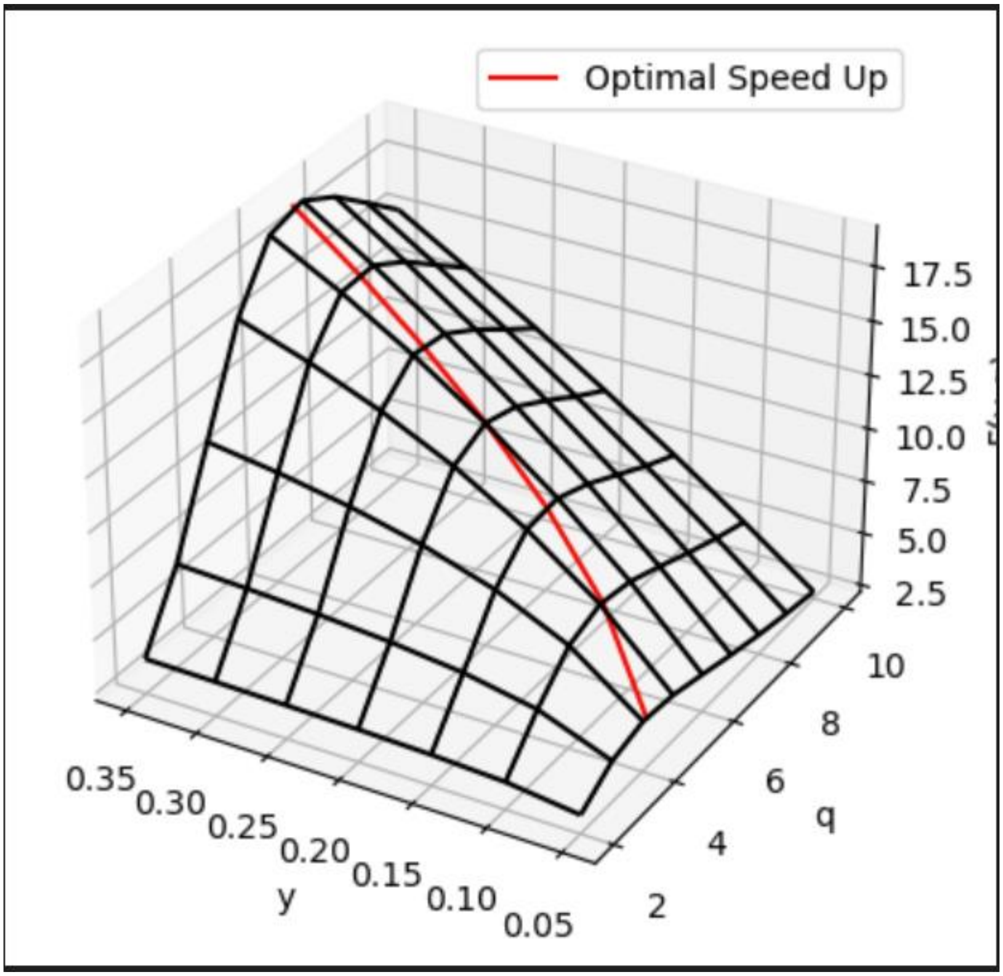
And then after calculating p(y) we get log and substitute in this equation F(y,q) = y * (n-1) / 2q + y (2^(n-q) – 1 + q) and get points of line which is the optimal solution and draw it .



```
y_values_line = y_values
result = (y_values * np.log(2) / (y_values + 2)) * n
result_1 = np.log(result) / np.log(2)
```



```
line_up = (y_values_line[1:] * (n - 1))
line_down = (2 * result_1[1:]) + y_values_line[1:] * (2 ** (10 - result_1[1:]) - 1 + result_1[1:])
line = line_up / line_down

# Plot the 3D curve
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_wireframe(y_values, x_values, z_values, color='black')

# Plot the line
ax.plot(y_values_line[1:], result_1[1:], line, color='red', label='Optimal Speed Up')

# Set labels for x, y, and z axes
ax.set_xlabel('y')
ax.set_ylabel('q')
ax.set_zlabel('F(y,q)')
ax.invert_xaxis()

# Add legend
ax.legend()

# Show the plot
plt.show()
```



Colab Link:

https://colab.research.google.com/drive/1jDmtIdUwhIUF-spIxm_PNwsTKVHLaXGF?usp=sharing