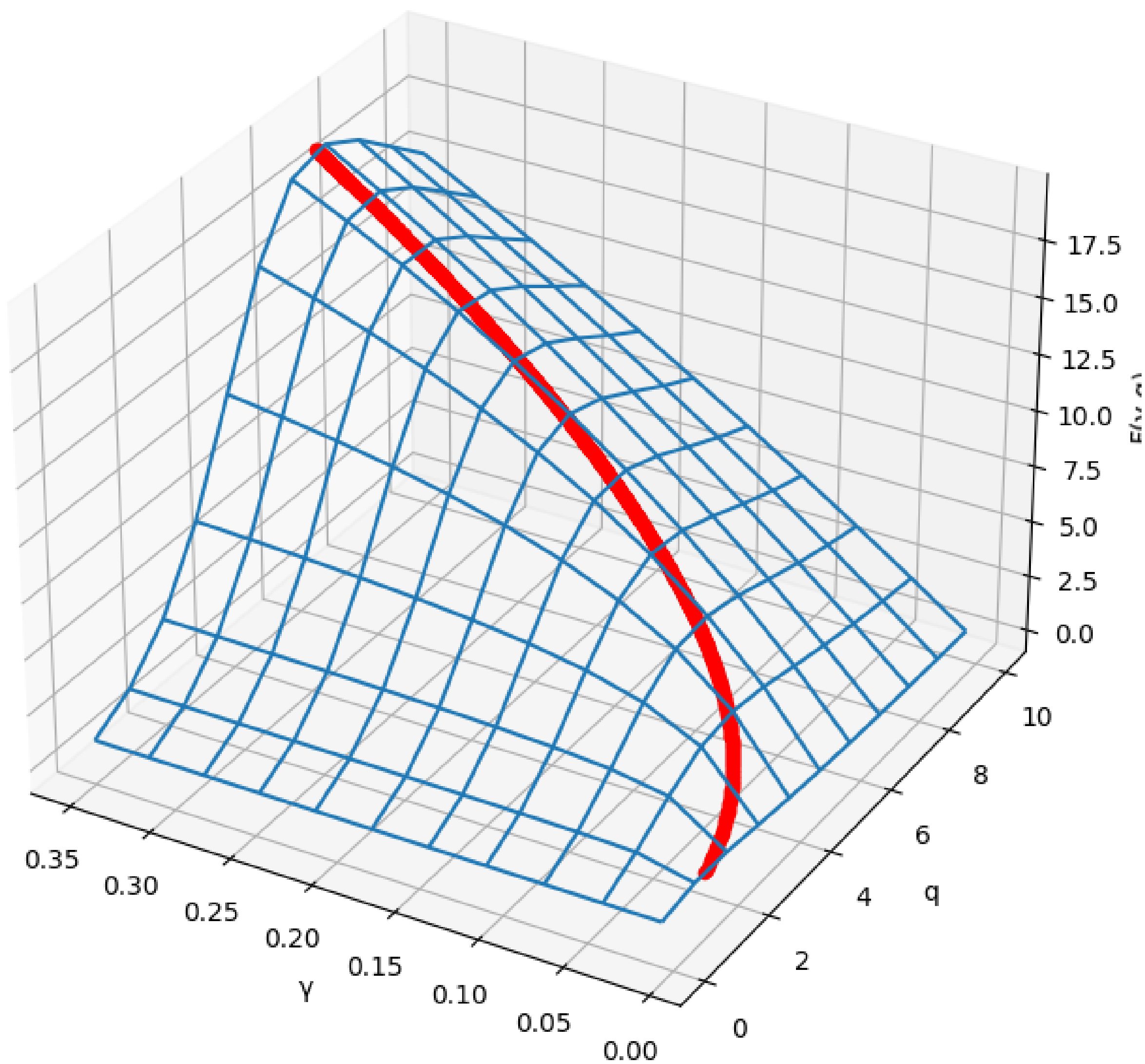
Gamal Abdel Hamid Nasef Nowesar
19015550

Result:



Explanation:

1. Generate array for gamma_values(0.007 → 0.35) & q_values(0 → 10)
2. Create the grid using gamma_values & q_values
3. Calculate speed_up using the formula:

$$S_\gamma(2^q, 2^k) = \frac{\gamma(2^k - 1)}{2q + \gamma(2^{k-q} - 1 + q)}$$

4. Calculate optimal_q_values using this formula (by taking log2 to the result):

$$p = \frac{\gamma \ln 2}{2+\gamma} n$$

5. Calculate optimal speed_up using optimal_q_values using the formula:

$$S_\gamma(2^q, 2^k) = \frac{\gamma(2^k - 1)}{2q + \gamma(2^{k-q} - 1 + q)}$$

6. Plot the wirefrome of the speedup
7. Draw the line of the optimal speedup using scatter
8. Show the plot

Code:

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

k = 10
n = 2 ** k
bins = 11

gamma_values = np.linspace(0.007, 0.35, bins)
q_values = np.linspace(0, 10, bins)
gamma, q = np.meshgrid(gamma_values, q_values)
speedup = gamma * (n - 1) / (2 * q + gamma * (2 ** (k - q) - 1 + q))

temp_gamma_values = np.linspace(0.007, 0.35, 1000)
```

```
optimal_q_values = np.log2((temp_gamma_values * np.log(2)) /
(temp_gamma_values + 2) * n)
optimal_speedups = temp_gamma_values * (n - 1) / (2 * optimal_q_values +
temp_gamma_values * (2 ** (k - optimal_q_values) - 1 + optimal_q_values))

fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
ax.invert_xaxis()
ax.plot_wireframe(gamma, q, speedup)

for i, q_val in enumerate(optimal_q_values):
ax.scatter(temp_gamma_values[i], q_val, optimal_speedups[i], color='red',
marker='o')

ax.set_xlabel('γ')
ax.set_ylabel('q')
ax.set_zlabel('F(γ,q)')

plt.show()
```

Colab link:
https://colab.research.google.com/drive/1SzCH8GSbtI1p8-E4rPBqkfqqrPqOeAFi?usp=sharing