import numpy as np  
import matplotlib.pyplot as plt  
  
# 已知条件  
AB\_length = 400 # AB的长度  
CD\_length = 10 # CD的长度  
OG\_height = 100 # OG的高度  
p = 50 # 抛物线的焦距  
  
# 抛物线方程 y = ax^2  
a = 1 / (4 \* p)  
  
# 反射镜数量和位置  
num\_mirrors = int(AB\_length / 2.5)  
mirror\_positions = np.linspace(-AB\_length / 2, AB\_length / 2, num\_mirrors + 1)  
  
# 模拟光线反射  
def simulate\_light\_reflection(mirror\_positions, a, p):  
 entering\_CD = 0 # 进入CD区间的光线数量  
 total\_lights = len(mirror\_positions) - 1 # 总光线数量，减1是因为两端的点不会放置反射镜  
  
 # 对于每个反射镜，模拟光线的反射  
 for i in range(1, len(mirror\_positions) - 1): # 忽略两端的点  
 x\_incident = mirror\_positions[i]  
 # 光线在抛物线上的反射点  
 y\_incident = a \* x\_incident \*\* 2 + OG\_height / 2  
  
 # 光线反射后的终点  
 x\_reflected = 0 # 抛物线对称，反射后光线将沿x轴  
 y\_reflected = 2 \* p - y\_incident + OG\_height / 2  
  
 # 检查反射光线是否进入CD区间  
 if -CD\_length / 2 <= x\_reflected <= CD\_length / 2 and 0 <= y\_reflected <= 2 \* OG\_height:  
 entering\_CD += 1  
  
 return entering\_CD, total\_lights  
  
# 执行模拟  
entering\_CD, total\_lights = simulate\_light\_reflection(mirror\_positions, a, p)  
  
# 计算比例  
ratio = entering\_CD / total\_lights  
  
# 输出结果  
print(f"Number of lights entering CD: {entering\_CD}")  
print(f"Total number of lights: {total\_lights}")  
print(f"Ratio of lights entering CD: {ratio:.2f}")  
  
# 绘制抛物线EF  
x\_vals = np.linspace(-AB\_length / 2, AB\_length / 2, 400)  
y\_vals = a \* x\_vals\*\*2 + OG\_height / 2  
plt.figure(figsize=(10, 5))  
plt.plot(x\_vals, y\_vals, label='Parabola EF')  
plt.title('Parabola EF for Light Reflection')  
plt.xlabel('Position along AB (m)')  
plt.ylabel('Height (m)')  
plt.legend()  
plt.grid(True)  
plt.show()