

```

import numpy as np
from scipy.optimize import minimize

# Define the channel matrix H
H = np.array([[3, 0, 8],
              [0, 1, 0],
              [4, 0, 6]])

# Transmit power (in dB)
P_dB = -1.75

# Convert transmit power from dB to linear scale
P = 10 ** (P_dB / 10)

# Receiver noise power (in dB)
sigma_n_squared_dB = 3

# Convert noise power from dB to linear scale
sigma_n_squared = 10 ** (sigma_n_squared_dB / 10)

# Function to maximize system capacity rate
def objective_function(x):
    SNR = np.abs(np.linalg.det(np.dot(H, np.diag(x)))) ** 2 / sigma_n_squared
    return -np.log2(1 + SNR)

# Initial guess for power allocation
x0 = np.array([P, 0, 0]) # Set the first transmit antenna to have all the power

# Define constraint for total transmit power
constraint = ({'type': 'eq', 'fun': lambda x: x[0] - P})

# Define bounds for power allocation (non-negativity constraint)
bounds = ((0, None), (0, None), (0, None))

# Solve the optimization problem
result = minimize(objective_function, x0, bounds=bounds, constraints=constraint)

# Extract the optimal power allocation
power_allocation = result.x
print("Optimal Power Allocation:", power_allocation)

```