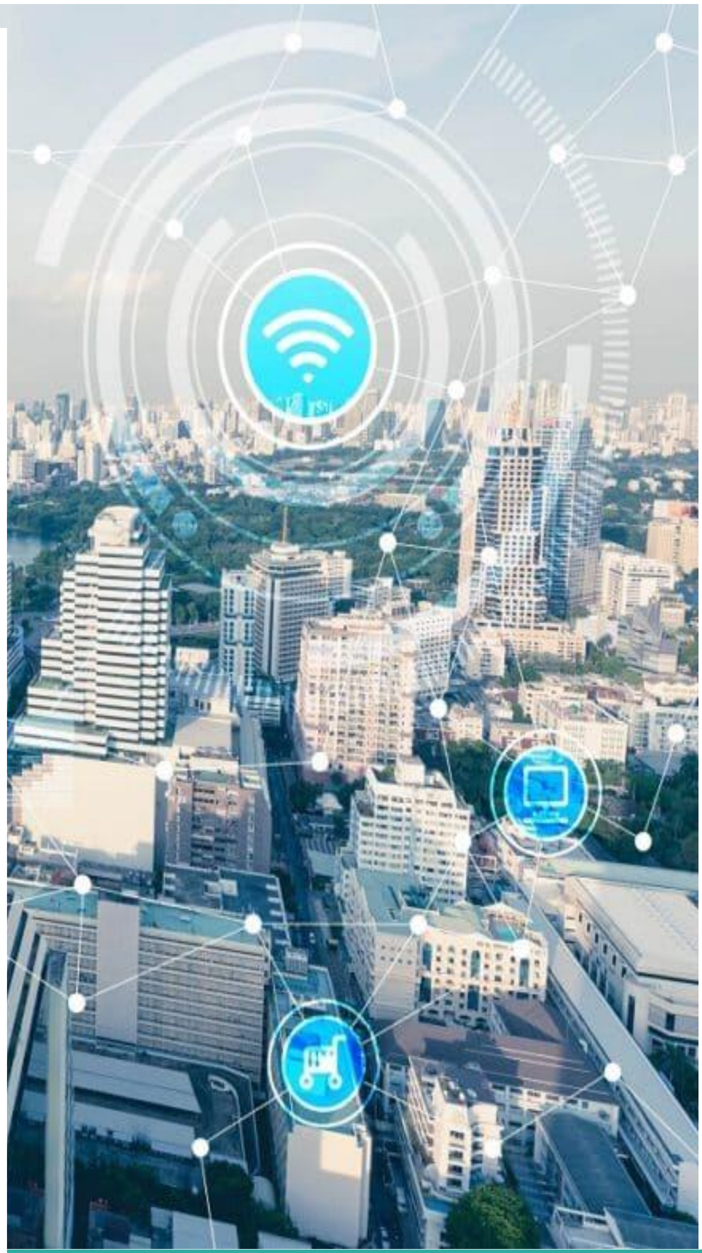
Information Theory Project



MAY 9

Team Members:

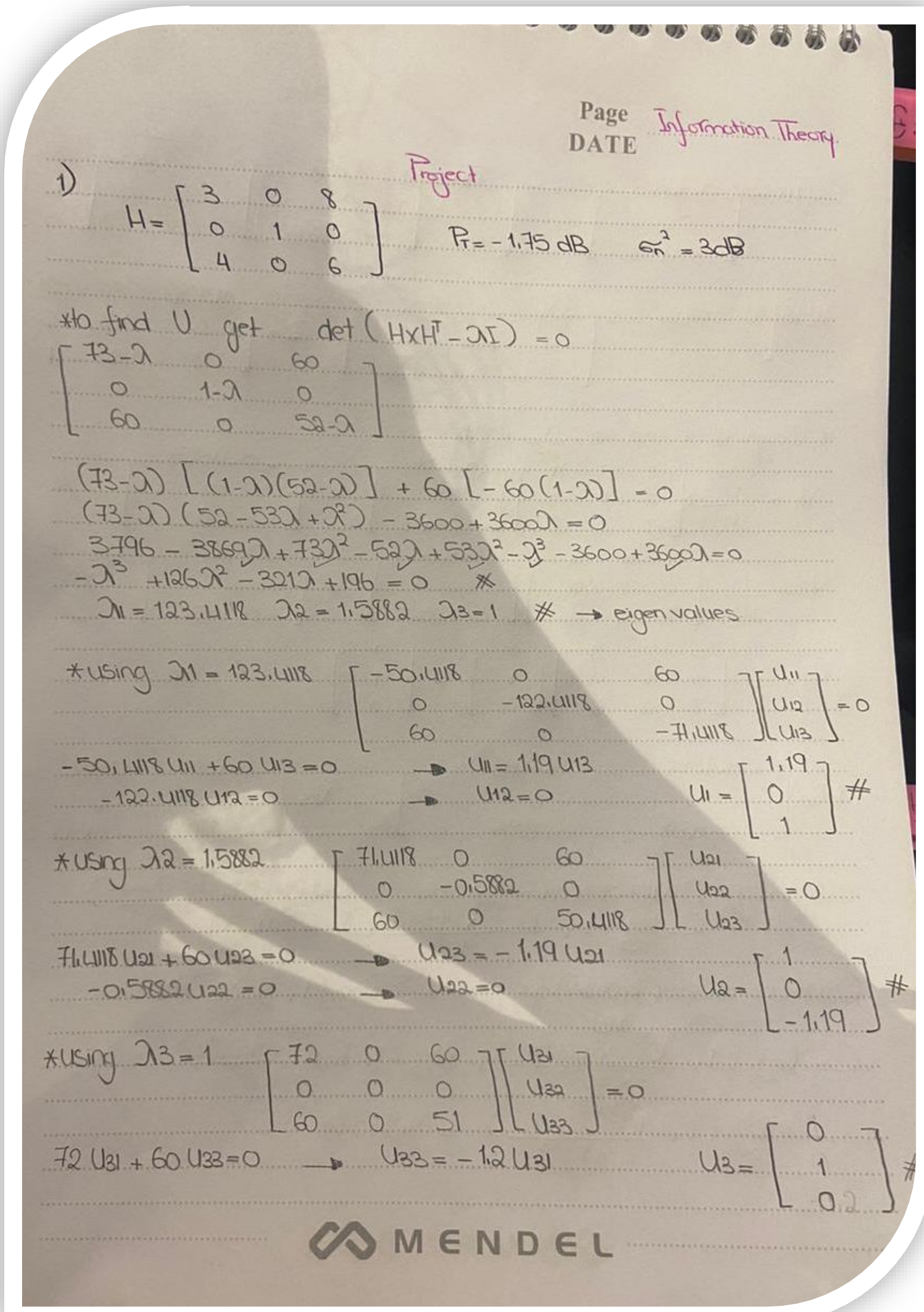
- Injy Nashaat Makram (49-6748)
- Kariman Akram Mostafa (49-2506)

Supervisor:

1. Dr. Engy Aly

Project Requirements:

1) Hand analysis:



Page
DATE
Information Theory

Project

1)

$$H = \begin{bmatrix} 3 & 0 & 8 \\ 0 & 1 & 0 \\ 4 & 0 & 6 \end{bmatrix} \qquad P_T = -1.75\ \text{dB} \qquad \sigma_n^2 = 3\text{dB}$$

*to find U get $\det(H \times H^T - \lambda I) = 0$

$$\begin{bmatrix} 73-\lambda & 0 & 60 \\ 0 & 1-\lambda & 0 \\ 60 & 0 & 52-\lambda \end{bmatrix}$$

$$(73-\lambda)[(1-\lambda)(52-\lambda)] + 60[-60(1-\lambda)] = 0$$

$$(73-\lambda)(52-53\lambda+\lambda^2) - 3600 + 3600\lambda = 0$$

$$3796 - 3869\lambda + 73\lambda^2 - 52\lambda + 53\lambda^2 - \lambda^3 - 3600 + 3600\lambda = 0$$

$$-\lambda^3 + 126\lambda^2 - 321\lambda + 196 = 0 \quad \#$$

$$\lambda_1 = 123.4118 \quad \lambda_2 = 1.5882 \quad \lambda_3 = 1 \quad \# \rightarrow \text{eigen values}$$

*using $\lambda_1 = 123.4118$

$$\begin{bmatrix} -50.4118 & 0 & 60 \\ 0 & -122.4118 & 0 \\ 60 & 0 & -71.4118 \end{bmatrix} \begin{bmatrix} U_{11} \\ U_{12} \\ U_{13} \end{bmatrix} = 0$$

$$-50.4118\,U_{11} + 60\,U_{13} = 0 \rightarrow U_{11} = 1.19\,U_{13}$$

$$-122.4118\,U_{12} = 0 \rightarrow U_{12} = 0$$

$$U_1 = \begin{bmatrix} 1.19 \\ 0 \\ 1 \end{bmatrix} \#$$

*using $\lambda_2 = 1.5882$

$$\begin{bmatrix} 71.4118 & 0 & 60 \\ 0 & -0.5882 & 0 \\ 60 & 0 & 50.4118 \end{bmatrix} \begin{bmatrix} U_{21} \\ U_{22} \\ U_{23} \end{bmatrix} = 0$$

$$71.4118\,U_{21} + 60\,U_{23} = 0 \rightarrow U_{23} = -1.19\,U_{21}$$

$$-0.5882\,U_{22} = 0 \rightarrow U_{22} = 0$$

$$U_2 = \begin{bmatrix} 1 \\ 0 \\ -1.19 \end{bmatrix} \#$$

*using $\lambda_3 = 1$

$$\begin{bmatrix} 72 & 0 & 60 \\ 0 & 0 & 0 \\ 60 & 0 & 51 \end{bmatrix} \begin{bmatrix} U_{31} \\ U_{32} \\ U_{33} \end{bmatrix} = 0$$

$$72\,U_{31} + 60\,U_{33} = 0 \rightarrow U_{33} = -1.2\,U_{31}$$

$$U_3 = \begin{bmatrix} 0 \\ 1 \\ 0.2 \end{bmatrix} \#$$

MENDEL

Page
DATE

$$\bar{U}_1 = \begin{bmatrix} -0.7656 \\ 0 \\ -0.6433 \end{bmatrix} \qquad \bar{U}_2 = \begin{bmatrix} -0.6433 \\ 0 \\ -0.7656 \end{bmatrix} \qquad \bar{U}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

* get D-matrix $\rightarrow \begin{bmatrix} \sqrt{\lambda_1} & 0 & 0 \\ 0 & \sqrt{\lambda_2} & 0 \\ 0 & 0 & \sqrt{\lambda_3} \end{bmatrix} \quad D = \begin{bmatrix} 11.109 & 0 & 0 \\ 0 & 1.26 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

* to get V & use $V_i = \frac{1}{\sigma_i} A^T u_i$

$$V_1 = \frac{1}{11.109} \begin{bmatrix} 3 & 0 & 4 \\ 0 & 1 & 0 \\ 8 & 0 & 6 \end{bmatrix} \begin{bmatrix} -0.7656 \\ 0 \\ -0.6433 \end{bmatrix} = \begin{bmatrix} -0.4383 \\ 0 \\ -0.8988 \end{bmatrix} \#$$

$$V_2 = \frac{1}{1.26} \begin{bmatrix} 3 & 0 & 4 \\ 0 & 1 & 0 \\ 8 & 0 & 6 \end{bmatrix} \begin{bmatrix} -0.6433 \\ 0 \\ 0.7656 \end{bmatrix} = \begin{bmatrix} 0.8988 \\ 0 \\ -0.4387 \end{bmatrix} \#$$

$$V_3 = \frac{1}{1} \begin{bmatrix} 3 & 0 & 4 \\ 0 & 1 & 0 \\ 8 & 0 & 6 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \#$$

$$U = \begin{bmatrix} -0.7656 & -0.6433 & 0 \\ 0 & 0 & 1 \\ -0.6433 & 0.7656 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} -0.4383 & 0.8988 & 0 \\ 0 & 0 & 1 \\ -0.8988 & -0.4387 & 0 \end{bmatrix}$$

multiplexing: ① with channel knowledge $C = \log_2\left(1 + \frac{P_T \lambda_{max}}{\sigma_n^2}\right)$

$$C = \log_2\left(1 + \frac{10^{-0.175} \times 123.4118}{10^{0.13}}\right) = 5.4039$$

② without channel knowledge $C = \sum \log_2\left(1 + \frac{P_T \lambda_i}{m \sigma_n^2}\right)$

$$C = \log_2\left(1 + \frac{10^{-0.175} \times 123.4118}{3 \times 10^{0.13}}\right) + \log_2\left(1 + \frac{10^{-0.175} \times 1.5882}{3 \times 10^{0.13}}\right)$$

$$+ \log_2\left(1 + \frac{10^{-0.175} \times 1}{3 \times 10^{0.13}}\right) = 4.27376$$

MENDEL

Page
DATE

diversity: 1 with channel knowledge $C = \log_2 \left(1 + \frac{P_T \lambda_{max}}{\sigma_n^2}\right)$

$$C = \log_2 \left(1 + \frac{10^{-0.175} \times 123.4118}{10^{0.13}}\right) = 5.4039$$

2 without channel knowledge $C = \log_2 \left(1 + \frac{P_T}{m\sigma_n^2} \Sigma \lambda_i\right)$

$$C = \log_2 \left(1 + \frac{10^{-0.175}}{3 \times 10^{0.13}} \left[123.4118 + 1.5882 + 1\right]\right) = 3.9135$$

2) Commented codes for Part I (b), (c) and (d):

- **Part I (b):** The result indicates that the majority of the available power is allocated to the first transmit antenna, with minimal power allocated to the other antennas. Specifically, the optimal power allocation is approximately 0.67 for the first antenna, while the remaining antennas receive negligible power.

- **Part I (c):** Optimal Transmit Powers: The optimal transmit powers are computed using the water-filling algorithm to maximize the MIMO channel capacity. These powers are allocated to each antenna based on the channel's eigenvalues, aiming to balance power across the antennas to achieve the highest capacity.

- **Part I (d):**

(i) Multiplexing with and without Channel Knowledge:

1. **With Channel Knowledge:** The multiplexing capacity, computed with full knowledge of the channel (i.e., eigenvalues), is approximately 5.404 bits/s/Hz. This capacity represents the maximum achievable data rate when the transmitter has knowledge of the channel's characteristics.
2. **Without Channel Knowledge:** In scenarios where the transmitter lacks channel knowledge, the multiplexing capacity decreases to approximately 4.274 bits/s/Hz. Without precise information about the channel, the capacity is reduced as the transmitter cannot fully exploit the channel's characteristics for optimal data transmission.

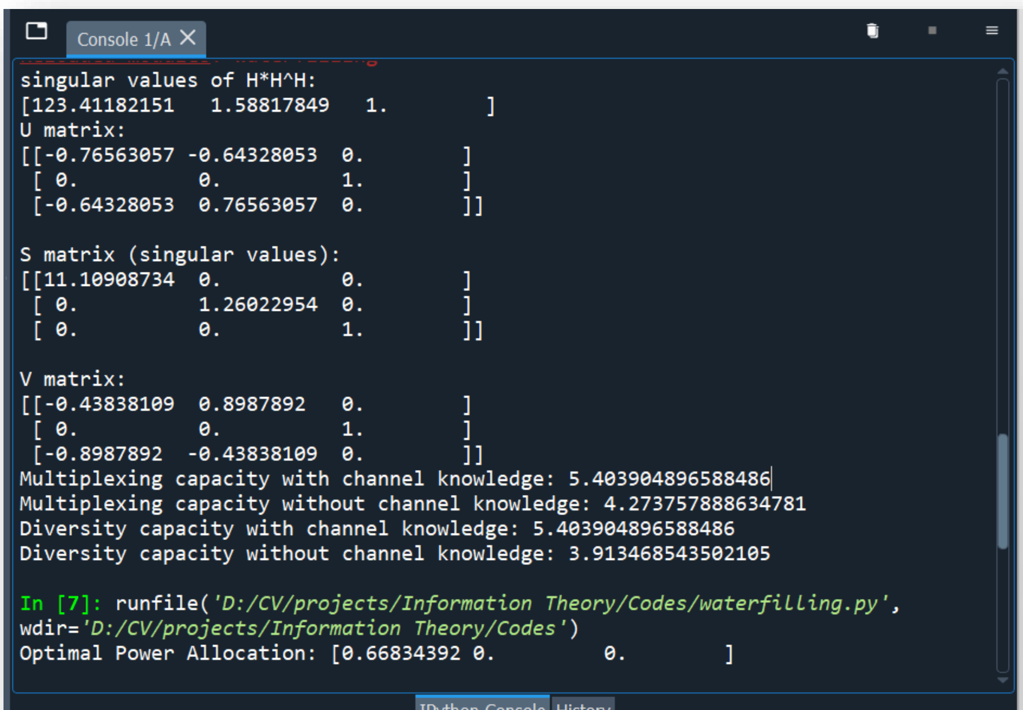
(ii) Diversity with and without Channel Knowledge:

1. **With Channel Knowledge:** The diversity capacity, calculated with knowledge of the channel, matches the multiplexing capacity at

approximately 5.404 bits/s/Hz. This indicates that with full channel knowledge, the system achieves its maximum diversity gain, resulting in the same capacity as multiplexing.

2. **Without Channel Knowledge:** In contrast, when the transmitter lacks channel knowledge, the diversity capacity decreases to approximately 3.913 bits/s/Hz. Without accurate information about the channel, the system's diversity gain is compromised, leading to a lower capacity compared to scenarios with channel knowledge.

3) Display of the results for Part I (b), (c) and (d):



```
Console 1/A
singular values of H*H^H:
[123.41182151   1.58817849   1.        ]
U matrix:
[[-0.76563057 -0.64328053  0.        ]
 [ 0.          0.          1.        ]
 [-0.64328053  0.76563057  0.        ]]

S matrix (singular values):
[[11.10908734  0.          0.        ]
 [ 0.          1.26022954  0.        ]
 [ 0.          0.          1.        ]]

V matrix:
[[-0.43838109  0.8987892   0.        ]
 [ 0.          0.          1.        ]
 [-0.8987892  -0.43838109  0.        ]]
Multiplexing capacity with channel knowledge: 5.403904896588486
Multiplexing capacity without channel knowledge: 4.273757888634781
Diversity capacity with channel knowledge: 5.403904896588486
Diversity capacity without channel knowledge: 3.913468543502105

In [7]: runfile('D:/CV/projects/Information Theory/Codes/waterfilling.py',
wdir='D:/CV/projects/Information Theory/Codes')
Optimal Power Allocation: [0.66834392 0.         0.        ]
IPython Console History
```

4) Brief introduction to the Rayleigh Fading channel in a MIMO system:

In a Multiple Input Multiple Output (MIMO) wireless communication system, the Rayleigh fading channel plays a pivotal role in understanding the behavior of signals as they propagate through the wireless medium.

The Rayleigh fading channel model is a statistical representation of the effects of multipath propagation in wireless communication environments. It assumes that the magnitude and phase of the received signal experience random variations due to the superposition of multiple reflected paths between the transmitter and receiver, each with different attenuation and phase shift characteristics. These variations are primarily caused by factors such as signal reflections, diffractions, and scattering from objects in the propagation environment.

One key characteristic of the Rayleigh fading channel is its statistical property, which follows a Rayleigh distribution for the envelope of the received signal. This means that the magnitude of the received signal varies randomly around a mean value, without any dominant line-of-sight component. Consequently, in Rayleigh fading channels, the received signal strength can experience deep fades, where the signal strength significantly drops, as well as peaks, where the signal strength momentarily increases.

In a MIMO system, where multiple antennas are used at both the transmitter and receiver, the Rayleigh fading channel model extends to encompass the spatial domain. Each antenna pair introduces additional paths for signal propagation, leading to spatial variations in the received signal strength and phase. Consequently, the capacity and performance of MIMO systems in Rayleigh fading channels are influenced by factors such as the number of antennas, the spacing between antennas, and the correlation between antennas.

5) Ergodic Rayleigh Fading channel capacity for a MIMO system:

The term "Ergodic" refers to the statistical property of the channel, where the capacity is calculated as the average capacity over many channel realizations. This statistical averaging accounts for the random fluctuations in the channel due to fading, providing a more realistic estimation of the system's performance.

The Rayleigh fading channel, as previously discussed, introduces random variations in signal strength and phase due to multipath propagation. In a MIMO system, these variations extend across multiple antenna pairs, resulting in spatially diverse channel conditions.

The capacity of the Ergodic Rayleigh Fading channel in a MIMO system is influenced by several factors, including the number of antennas at both the transmitter and receiver, the signal-to-noise ratio (SNR) of the communication link, and the correlation between the antennas.

To calculate the Ergodic Rayleigh Fading channel capacity, one typically employs mathematical tools such as singular value decomposition (SVD) to analyze the channel matrix and determine the singular values representing the channel's characteristics. These singular values capture the spatial diversity of the channel and play a crucial role in determining the system's capacity.

By averaging the capacity over numerous channel realizations, the Ergodic Rayleigh Fading channel capacity provides valuable insights into the expected performance of the MIMO system in practical scenarios affected by fading. This statistical approach enables engineers to design communication systems with robustness to fading effects and optimal utilization of spatial diversity for achieving higher data rates and improved reliability.

In summary, the Ergodic Rayleigh Fading channel capacity in a MIMO system quantifies the maximum achievable data rate considering the statistical nature of fading effects, offering a realistic measure of system performance under real-world conditions.

6) Rules used to calculate the Ergodic capacity in a MIMO system:

1. **Channel Matrix (H): The channel matrix H represents the linear transformation between the transmitted signals and the received signals in the MIMO system. It accounts for fading effects between the transmit and receive antennas. This matrix is given in the project and serves as the basis for further analysis.**

2. **Singular Value Decomposition (SVD): The channel matrix H is decomposed using SVD to obtain its singular value decomposition. SVD expresses H as the product of three matrices: U, D, and V. Here, U and V are unitary matrices, and D is a diagonal matrix containing the singular values of H. These singular values capture the magnitudes of the channel gains along each transmission direction.**

3. **Number of Realizations: The number of realizations is defined to perform the averaging of the capacity over these realizations. This averaging accounts for the statistical variations in the channel due to fading effects. A larger number of realizations provides a more accurate estimate of the system's performance.**

4. **Capacity Formula: The capacity formula used to calculate the Ergodic capacity depends on the MIMO system model employed:**

   Multiplexing:

   - **Sending a Different Symbol on each Orthogonal Channel**
   - **Main target is increasing data rate**
   - **Given $p_i = E[s_i^2]$ is power allocated on channel i:**

(i) With Knowledge: Capacity = $log_2(1 + \frac{P_{T^*} \lambda_{max}}{\sigma_n^2})$

(ii) Without Knowledge: Capacity = $\sum_{i=1}^{k} log_2(1 + \frac{P_{T^*} \lambda i}{M * \sigma_n^2})$

Diversity:

- Sending the same Symbol on all Orthogonal Channels
- Main target is enhancing SNR (and in turn data rate as well)
- Given p_i is power allocated on channel i for each symbol s:

(i) With Knowledge: Capacity = $log_2(1 + \frac{P_{T^*} \lambda_{max}}{\sigma_n^2})$

(ii) Without Knowledge: Capacity = $log_2(1 + \frac{P_T}{\sigma_n^2} \sum_{i=1}^{k} \lambda_i)$

7) Commented codes for Part II (a) and (b):

2. Part II (a):

Antennas=1:

[0.84, 0.99, 1.17, 1.32, 1.52, 1.73, 1.89, 2.21, 2.37, 2.66, 2.90, 3.22, 3.43, 3.74, 3.98, 4.31, 4.51, 5.00, 5.20, 5.48, 5.96]

Antennas=2:

[1.72, 1.96, 2.24, 2.55, 2.88, 3.28, 3.66, 4.13, 4.56, 5.09, 5.54, 6.06, 6.64, 7.11, 7.81, 8.24, 8.93, 9.41, 10.06, 10.53, 11.26]

Antennas=4:

[3.38, 3.85, 4.48, 5.18, 5.80, 6.52, 7.34, 8.22, 9.09, 10.01, 11.00, 11.93, 12.99, 13.99, 15.16, 16.34, 17.28, 18.59, 19.77, 20.94, 22.15]

Antennas=8:
[6.72, 7.79, 8.94, 10.20, 11.59, 13.05, 14.68, 16.38, 18.08, 19.96, 21.80, 23.73, 25.81, 27.89, 30.15, 32.28, 34.54, 36.80, 39.19, 41.42, 44.02]

Antennas=16:
[13.40, 15.55, 17.83, 20.42, 23.24, 26.18, 29.33, 32.59, 36.07, 39.83, 43.49, 47.56, 51.56, 55.76, 60.07, 64.48, 68.95, 73.53, 78.33, 83.01, 87.90]

Antennas=32:
[26.80, 31.06, 35.78, 40.88, 46.41, 52.25, 58.58, 65.17, 72.27, 79.58, 87.15, 95.07, 103.16, 111.61, 120.23, 128.98, 137.96, 147.15, 156.47, 165.92, 175.44]

Antenna Increase Boosts Capacity: **Adding antennas improves MIMO system capacity, indicating the significance of spatial diversity in enhancing communication performance.**

SNR Directly Impacts Capacity: **Higher SNR levels lead to increased capacity, underlining the importance of managing noise relative to signal strength for better communication.**

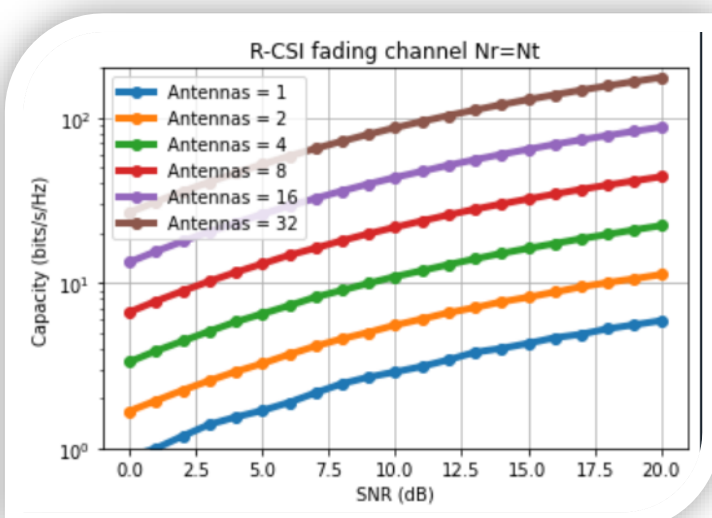
Diminishing Returns with Antennas: **Adding more antennas improves the capacity of the MIMO system. However, the rate at which capacity improves slows down as more antennas are added. This means that while extra antennas still increase capacity, the increase becomes less significant with each additional antenna. This idea essentially captures the notion that the benefit gained from adding more antennas diminishes as you keep adding more.**

```
Console 1/A

In [3]: runfile('C:/Users/20109/Downloads/untitled2.py', wdir='C:/Users/20109/Downloads')
Antennas=1:  [0.8431094300704797, 0.9880704209646417, 1.170520738702203, 1.3210342521093925, 1.5200257327144506, 1.7308505467137498,
1.8938718857562888, 2.2059854967039483, 2.3704174384419456, 2.6584677340599363, 2.904851048984934, 3.2225076126679286, 3.438841470057697,
3.74220309817985, 3.9797522455580574, 4.306072533030694, 4.514279725978504, 4.997615329641366, 5.201025411392528, 5.479216854419082,
5.964723698761352]
Antennas=2:  [1.72107238426706, 1.961084281800318, 2.2361875906267796, 2.5480329814205795, 2.8806298924711946, 3.284202490070174З,
3.655519606962855, 4.128776707998406, 4.560765200018817, 5.08646160151559, 5.539833476976671, 6.0559445564898535, 6.637615012885507,
7.10697690942657, 7.807243298441643, 8.236631884248455, 8.930116534543165, 9.414240262809729, 10.056703538576162, 10.529895101811453,
11.261388162072315]
Antennas=4:  [3.379197737531572, 3.8521825079918597, 4.477391468084756, 5.179953613390631, 5.804749607706829, 6.520839251592926,
7.342572796834849, 8.219773565488547, 9.090228204203653, 10.006219712976993, 10.999041009033258, 11.934398501555336, 12.992619453290983,
13.989379678301713, 15.158714852658091, 16.339982980924656, 17.282657806738026, 18.592901900830558, 19.765490912089213, 20.937534200321924,
22.152638170514408]
Antennas=8:  [6.723057446638721, 7.794691514681363, 8.935985377178607, 10.199474852905261, 11.590326791425309, 13.048481867908055,
14.67602726520352, 16.377802925204893, 18.076148474378474, 19.955605246654752, 21.795511699969296, 23.733875288582208, 25.807695792139466,
27.889436107122197, 30.14858396956646, 32.2782789410469, 34.54169642967776, 36.80355449475424, 39.19126124459985, 41.41752815402136,
44.01621909134463]
Antennas=16:  [13.400949353152248, 15.547581487137755, 17.828510939659566, 20.422718549631853, 23.235905886542184, 26.176950791309544,
29.332166132805703, 32.59000368376458, 36.070188993713835, 39.83226338477552, 43.49487256710837, 47.55835076554954, 51.56178553100341,
55.75713744787547, 60.06702647345429, 64.47516461288502, 68.94619830524995, 73.53453006703371, 78.33107627149138, 83.01431789555727,
87.89649330234545]
Antennas=32:  [26.804367927841437, 31.062405199432302, 35.778584679886706, 40.878937124332204, 46.41324478995248, 52.25270062729868,
58.580849541260356, 65.17459118662353, 72.27102754227035, 79.57879927133533, 87.15052039217466, 95.0692902327523, 103.16173104842075,
111.6115797224618, 120.23400329603678, 128.97851680671164, 137.95878228190995, 147.14778545718258, 156.4748346206515, 165.92373075272926,
175.43890224374118]

IPython Console  History
custom ()  Completions: custom  ✓ LSP: Python  Line 46, Col 1  UTF-8  CRLF  RW  Mem 74%
```

3. **Part II (b): The plot shows the channel capacity (bits/s/Hz) versus SNR (dB) for different numbers of antennas. As expected, the channel capacity increases with both SNR and the number of antennas. This is because more antennas provide more spatial diversity, which can help to mitigate the effects of fading.**



```python
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)
```

```python
import numpy as np


# Given channel matrix H
m=3;
H = np.array([[3, 0, 8],
              [0, 1, 0],
              [4, 0, 6]])

# Calculate eigenvalues of H*H^H
singularvalues = np.linalg.eigvals(H @ H.T)

# Print the singular values
print("singular values of H*H^H:")
print(singularvalues)




def svd_channel_matrix(H):
    U, D, Vh = np.linalg.svd(H)
    return U, D, Vh.T  # Transpose Vh to match MATLAB's convention

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

# Call the function
U, D, V = svd_channel_matrix(H)

# Print the results
print("U matrix:")
print(U)
print("\nS matrix (singular values):")
print(np.diag(D))
print("\nV matrix:")
print(V)




def compute_capacity_multiplexing(H, sigma_n_squared, P_total, channel_knowledge=True):
    U, s, Vh = svd_channel_matrix(H)
    if channel_knowledge:
        capacity = (np.log2(1 + P_total*(max(singularvalues)) / sigma_n_squared))
    else:
        z=P_total*singularvalues
        o=m*sigma_n_squared
        capacity = np.sum(np.log2(1 + z/o))
    return capacity

def compute_capacity_diversity(H, sigma_n_squared, P_total, channel_knowledge=True):
    U, s, Vh = svd_channel_matrix(H)
    if channel_knowledge:

        capacity = (np.log2(1 + P_total*(max(singularvalues)) / sigma_n_squared))
    else:

        f=np.sum(singularvalues)
        x=P_total*f
        y=m*sigma_n_squared

        capacity = (np.log2(1 + x/y))
    return capacity

# Example usage
H = np.array([[3, 0, 8], [0, 1, 0], [4, 0, 6]])
sigma_n_squared = 10**(3/10)  # Convert dB to linear scale
P_total = 10**((-1.75)/10)    # Convert dB to linear scale

# Multiplexing with channel knowledge
capacity_multiplexing_with_ck = compute_capacity_multiplexing(H, sigma_n_squared, P_total)
print("Multiplexing capacity with channel knowledge:", capacity_multiplexing_with_ck)

# Multiplexing without channel knowledge
capacity_multiplexing_without_ck = compute_capacity_multiplexing(H, sigma_n_squared, P_total, channel_knowledge=False)
```

```python
print("Multiplexing capacity without channel knowledge:", capacity_multiplexing_without_ck)

# Diversity with channel knowledge
capacity_diversity_with_ck = compute_capacity_diversity(H, sigma_n_squared, P_total)
print("Diversity capacity with channel knowledge:", capacity_diversity_with_ck)

# Diversity without channel knowledge
capacity_diversity_without_ck = compute_capacity_diversity(H, sigma_n_squared, P_total, channel_knowledge=False)
print("Diversity capacity without channel knowledge:", capacity_diversity_without_ck)
```

```python
import numpy as np

No_of_antennas = [1, 2, 4, 8, 16, 32]

# The number of blocks for ergodic channel (realizations)
No_of_blocks_ergodic = 1000

# SNR values in dB
SNR_range = np.arange(0, 21)

# Convert SNR values from dB to linear
SNR = 10 ** (SNR_range / 10)

# Function to calculate capacity for a given SNR and number of antennas
def calculate_capacity(No_of_transmitter, snr):
    # Set number of receivers equal to number of transmitters
    No_of_reciever = No_of_transmitter

    # Generate channel matrix H
    H = (1 / np.sqrt(2)) * (np.random.randn(No_of_transmitter, No_of_reciever, No_of_blocks_ergodic) + 1j * np.random.randn(No_of_transmitter, No_of_reciever, No_of_blocks_ergodic))

    # Initialize array to store channel capacities for each block
    Channel_Capacity_Block = np.zeros(No_of_blocks_ergodic)

    # Loop over each block
    for k in range(No_of_blocks_ergodic):
        # Extract H for current block
        Hcurr = H[:, :, k]

        # Compute singular values of H
        singular_values = np.linalg.svd(Hcurr, compute_uv=False)

        # Calculate eigenvalues from squared singular values
        eigenvalues = singular_values ** 2

        # Calculate channel capacity for each block
        Channel_Capacity_Block[k] = np.sum(np.log2(1 + (snr * eigenvalues) / No_of_transmitter))

    # Calculate mean channel capacity
    return np.mean(Channel_Capacity_Block)

# Calculate Ergodic Rayleigh Fading channel capacity for each configuration
for No_of_transmitter in No_of_antennas:
    capacities = [calculate_capacity(No_of_transmitter, snr) for snr in SNR]
    print(f"Antennas={No_of_transmitter}: ", capacities)
```