1. If a cellular mobile telephone system uses two 25khz simplex channel to provide full duplex Voice and control channel and having a total 36Mhz of bandwidth. Compute the number of Channels available/channel. If it uses 4 cell reuse. Determine equitable distribution of control channel and voice channel if the system has k=4 and 1Mhz of the allocated spectrum is dedicated to control channels. (Lab 1, Q 1)

```
BW=(int(input("Enter the total Bandwidth of channel in MHz : ")))*(10**6)
simplex_chan_bw=int(input("Enter the Simplex Channel Bandwidth in KHz : "))
duplex_chan=simplex_chan_bw*2
print("Duplex Channel Bandwidth is : {} kHz/channel".format(duplex_chan))
tot_chan=np.round(BW/(duplex_chan*(10**3)),0)
print("Total availabe channels are : {} channels".format(tot_chan))
```

```
Enter the total Bandwidth of channel in MHz : 36
Enter the Simplex Channel Bandwidth in KHz : 25
Duplex Channel Bandwidth is : 50 kHz/channel
Total availabe channels are : 720.0 channels
```

In [4]:

```
n=int(input("Enter cell reuse pattern (4,7 or 12) : "))
avail_chan=math.ceil(tot_chan/n)
print("Number of channels available per cell: {}".format(avail_chan))
#control channels
spectrum=int(input("Enter the frequency spectrum for control channel in MHz:"))*(10**6)
S=math.ceil(spectrum/(duplex_chan*(10**3)))
print("Number of control channels out of {} available channels are {}".format(tot_chan,S))
#Number of control channel per cells
con_chan=math.ceil(S/n)
print("Control channels for {} reusable cells are : {}".format(n,con_chan))
voice_chan=avail_chan-con_chan
print("Voice channels for {} reusable cells are : {}".format(n,voice_chan))
```

```
Enter cell reuse pattern (4,7 or 12) : 4
Number of channels available per cell: 180
Enter the frequency spectrum for control channel in MHz:1
Number of control channels out of 720.0 available channels are 20
Control channels for 4 reusable cells are : 5
Voice channels for 4 reusable cells are : 175
```

2. Consider geographical area of a cellular system is 480Km2. A total of 910 radio channels are available for traffic handling suppose, area of a cell is 8 Km2. How many times would the cluster size of 7 have to be replicated in order to cover the entire service area? Calculate the system capacity. If the cluster size is decreased from 7 to 4, comment on results (Lab 1, Q2)

```
area = int(input("Enter the geographical area of the cell in square kilometers : "))
cov_area = int(input("Enter the coverage area of the cell in square kilometers : "))
radio_chan = int(input("Enter number of radio channels available for traffic handling : "))
```

```
Enter the geographical area of the cell in square kilometers : 480
Enter the coverage area of the cell in square kilometers : 8
Enter number of radio channels available for traffic handling : 910
```

In [8]:

```
k=[]
sys_cap=[]
print("Enter the cluster sizes")
for i in range(5):
    x=int(input())
    k.append(x)
    i=i+1
print("The cluster sizes are {}".format(k))
```

```
Enter the cluster sizes
1
3
4
7
9
The cluster sizes are [1, 3, 4, 7, 9]
```

```
for clust in k:
    clust_cov = clust * cov_area
    print("Total coverage area of the cluster is : {}".format(clust_cov))
    cell_area = math.ceil(area/clust_cov)
    print("Number of times the cluster has to be replicated to cover the entire service are
    is {}".format(cell_area))
    cell_cap = math.ceil(radio_chan/clust)
    print("Cell Capacity is {} channels/cell".format(cell_cap))
    sys = round(radio_chan * cell_area,0)
    print("System capacity is {} channels\n".format(sys))
    sys_cap.append(sys)
print("The system capacities for cluster sizes {} are {}".format(k,sys_cap))
```

```
Total coverage area of the cluster is : 32
Number of times the cluster has to be replicated to cover the entire service
area of cellular system     is 15
Cell Capacity is 228 channels/cell
System capacity is 13650 channels

Total coverage area of the cluster is : 56
Number of times the cluster has to be replicated to cover the entire service
area of cellular system     is 9
Cell Capacity is 130 channels/cell
System capacity is 8190 channels
```

```
The system capacities for cluster sizes [1, 3, 4, 7, 9] are [54600, 18200, 1
3650, 8190, 6370]
```

3. If a signal to interference ratio of 15dB is required for satisfactory forward channel performance of a cellular system, calculate frequency reuse factor and cluster size that should be used for maximum capacity if path loss exponent is η=3 and η =4. Assume cluster size N = 4 and 7. Comment on results. (Lab 2, Q1)

```
N=[3,4,7,9,12,13]
SI_plt=[]
si=15 #S/I in dB
SI=10**(si/10)
n=int(input("Enter the value of Path Loss Exponent:"))
for i in N :
    io=(i-1)
    Q=round(np.sqrt(3*i),2)
    print("The Co-channel Reuse Ratio(Q) is {}".format(Q))
    SI_new=round((Q**n)/io,2)
    SIdB=round(10*np.log10(SI_new),2)
    print("The Value of S/I for cluster size {} is {} dB".format(i,SIdB))
    SI_plt.append(SIdB)
    if SIdB>si :
        print("The design with Path Loss Exponent {} is accepted \n".format(n))
    else :
        print("The design with Path Loss Exponent {} is rejected \n".format(n))
print(SI_plt)
```

```
The Co-channel Reuse Ratio(Q) is 3.46
The Value of S/I for cluster size 4 is 11.4 dB
The design with Path Loss Exponent 3 is rejected

The Co-channel Reuse Ratio(Q) is 4.58
The Value of S/I for cluster size 7 is 12.04 dB
The design with Path Loss Exponent 3 is rejected
```

```
SI_plt2=[]
si=15 #S/I in dB
SI=10**(si/10)
n=int(input("Enter the value of Path Loss Exponent:"))
for i in N :
    io=(i-1)
    Q=round(np.sqrt(3*i),2)
    print("The Co-channel Reuse Ratio(Q) is {}".format(Q))
    SI_new=round((Q**n)/io,2)
    SIdB=round(10*np.log10(SI_new),2)
    print("The Value of S/I for cluster size {} is {} dB".format(i,SIdB))
    SI_plt2.append(SIdB)
    if SIdB>si :
        print("The design with Path Loss Exponent {} is accepted \n".format(n))
    else :
        print("The design with Path Loss Exponent {} is rejected \n".format(n))
print(SI_plt2)
```

```
The Co-channel Reuse Ratio(Q) is 3.46
The Value of S/I for cluster size 4 is 16.79 dB
The design with Path Loss Exponent 4 is accepted

The Co-channel Reuse Ratio(Q) is 4.58
The Value of S/I for cluster size 7 is 18.65 dB
The design with Path Loss Exponent 4 is accepted
```

4. A cellular system has S/I ratio of i) 18 dB ii) 20 dB. The frequency reuse factor is N=7. Calculate the worst case for signal to co-channel interference ratio. Are the designs acceptable or rejected? Assume path loss exponent as 4. Comment on the results. (Lab 2, Q 2&3)

```
pth_loss = int(input("Enter the value of Path Loss Exponent : "))
freq_reuse = int(input("Enter the frequency reuse factor : "))
Q = round(((3*freq_reuse)**0.5),2)
print("The value of Co-channel Reuse Ratio for N = {} is {}".format(freq_reuse,Q))
S = ((Q**pth_loss)/6)
S_dB = round(10*np.log10(S),2)
print("S/I ratio is {} dB".format(S_dB))
if(S_dB > 18):
    print("The design is accepted for Path Loss Exponent n = {}".format(pth_loss))
else:
    print("The design is rejected for Path Loss Exponent n = {}".format(pth_loss))
```

```
Enter the value of Path Loss Exponent : 4
Enter the frequency reuse factor : 7
The value of Co-channel Reuse Ratio for N = 7 is 4.58
S/I ratio is 18.65 dB
The design is accepted for Path Loss Exponent n = 4
```

Q.3) If the acceptable S/I is now 20 dB, will the cluster size determined in problem 2 be adequate? If not, then what should be the cluster size?

In [6]:

```
pth_loss = int(input("Enter the value of Path Loss Exponent : "))
freq_reuse = int(input("Enter the frequency reuse factor : "))
Q = round(((3*freq_reuse)**0.5),2)
print("The value of Co-channel Reuse Ratio for N = {} is {}".format(freq_reuse,Q))
S = ((Q**pth_loss)/6)
S_dB = round(10*np.log10(S),2)
print("S/I ratio is {} dB".format(S_dB))
if(S_dB > 20):
    print("The design is accepted for Path Loss Exponent n = {}".format(pth_loss))
else:
    print("The design is rejected for Path Loss Exponent n = {}".format(pth_loss))
```

```
Enter the value of Path Loss Exponent : 4
Enter the frequency reuse factor : 7
The value of Co-channel Reuse Ratio for N = 7 is 4.58
S/I ratio is 18.65 dB
The design is rejected for Path Loss Exponent n = 4
```

5. A hexagonal cell within a 4-cell system has a radius of 1.387 km. A total of 60 channels are used within the entire system. If the load per user is 0.029 Erlangs, and λ = 1call / hour, compute the following for an Erlang C system that has a 5% probability of a delayed call: (a) How many users per square kilometer will this system support? (a) What is the probability that a delayed call will have to wait for more than 10s? (c) What is the probability that a call will be delayed for more than 10 seconds? (Given Traffic Intensity A = 9 from Erlang C chart) (Lab 3, Q2)

```
R = float(input("Enter cell radius (in km) : "))
area = np.round(2.598*(R**2),0)
print("Area covered per cell is {} sq.km".format(area))
N = int(input("Enter number of cells in a cluster : "))
total_chans = int(input("Enter total number of channels : "))
C = total_chans/N
A = 9          #From Erlang C chart, for C = 15
Au = 0.029
lam = 1        #call/hour
t = 10         #seconds
delayed_prob = 0.05     #Probability of delayed call
users_per_sqkm = np.round(((A/Au)/area),2)

print()
print("Number of users per sq. km are {}\n".format(users_per_sqkm))
hold_time = (Au*3600)/lam
print("Holding time is {} seconds\n".format(hold_time))
hold_prob = np.round((math.exp(-(C-A)*t/hold_time)),5)
new_prob= np.round((hold_prob*100),3)
delay_prob = np.round((hold_prob*delayed_prob),5)
new_prob1 = delay_prob*100
print("Probability that a delayed call will have to wait for more \
          than 10 seconds is {} %\n".format(new_prob))
print("Probability that a call is delayed more than 10 seconds \
          is {} % \n".format(new_prob1))
```

```
Enter cell radius (in km) : 1.387
Area covered per cell is 5.0 sq.km
Enter number of cells in a cluster : 4
Enter total number of channels : 60

Number of users per sq. km are 62.07

Holding time is 104.4 seconds

Probability that a delayed call will have to wait for more than 10 seconds i
s 56.287 %

Probability that a call is delayed more than 10 seconds is 2.814 %
```

6. Determine (a) The channel capacity for a cellular system service area comprised of seven macrocell with 16 channels per macrocell (b) Channel capacity if each macrocell is split into four minicells (c) Channel capacity if each minicell is further split into four microcells. Comment on results. (Lab 4, Q2)

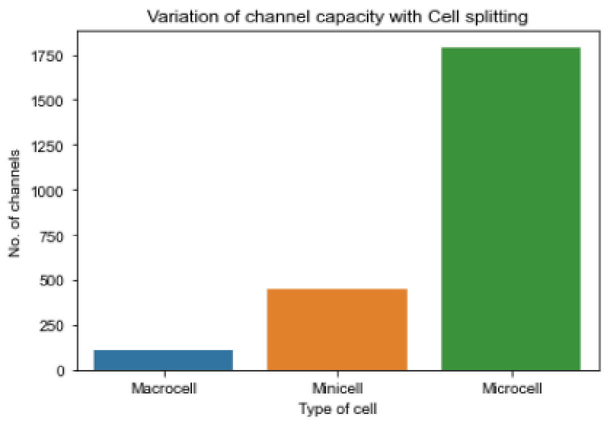
```
#a) Channel capacity for macrocell configuration
macro_per_sys=int(input("Enter the number of Macrocells per system : "))
macro_chan=int(input("Enter the number of channels per macrocell : "))
macro_sys_chan=macro_per_sys*macro_chan
print(f"The channel capacity of the Macrocell system is {macro_sys_chan} channels")

#b) Channel capacity for minicell configuration
mini_per_macro=int(input("Enter the number of minicells per macrocell : "))
mini_chans=macro_sys_chan*mini_per_macro
print(f"Channel capacity of Minicell system is {mini_chans} channels")

#c) Channel capacity for microcell configuration
micro_per_mini=int(input("Enter the number of microcells per minicell : "))
micro_chans=mini_chans*micro_per_mini
print(f"Channel capacity of Microcell system is {micro_chans} channels")
```

```
Enter the number of Macrocells per system : 7
Enter the number of channels per macrocell : 16
The channel capacity of the Macrocell system is 112 channels
Enter the number of minicells per macrocell : 4
Channel capacity of Minicell system is 448 channels
Enter the number of microcells per minicell : 4
Channel capacity of Microcell system is 1792 channels
```

```
chans=[macro_sys_chan,mini_chans,micro_chans]
cells=['Macrocell','Minicell','Microcell']
sns.barplot(x=cells,y=chans)
sns.set_style('whitegrid')
plt.title("Variation of channel capacity with Cell splitting")
plt.xlabel("Type of cell")
plt.ylabel("No. of channels")
```

Out[4]:

```
Text(0, 0.5, 'No. of channels')
```



7. Consider a cellular system with a radius of 2 Km which is split into smaller cells with a radius of 1 Km. Let each cell site be assigned 120 channels regardless of the cell size. How many times will the number of channels contained in a 6X6 Km2 area centered around small cell 'S' be increased with cell splitting as compared to without cell splitting? (Lab 4,Q3)

```
no_lcells=int(input("Enter the number of Large cells in the given area (visually) : "))
chans_per_lcell=int(input("Enter the number of channels per large cell : "))
nosplit_chans=no_lcells*chans_per_lcell
print(f"Number of channels without splitting are {nosplit_chans}")

lcell_rad=int(input("Enter the radius of the large cell (in km) : "))
splitcell_rad=int(input("Enter the radius of the split cell (in km) : "))
splitcells_in_sq= (lcell_rad/splitcell_rad)**2 * no_lcells
print(f"Theoretical number of split cells within the square are {splitcells_in_sq:.0f}")

act_splitcells_in_sq=splitcells_in_sq-1
chans_per_split=chans_per_lcell    #Channels per large cell is equal to channels per split
split_chans=act_splitcells_in_sq*chans_per_split
print(f"Number of channels with cell splitting are {split_chans:.0f}")

chans_increase=split_chans/nosplit_chans
print(f"Increase in the number of channels is {chans_increase} times")
```

```
Enter the number of Large cells in the given area (visually) : 4
Enter the number of channels per large cell : 120
Number of channels without splitting are 480
Enter the radius of the large cell (in km) : 2
Enter the radius of the split cell (in km) : 1
Theoretical number of split cells within the square are 16
Number of channels with cell splitting are 1800
Increase in the number of channels is 3.75 times
```

8. A cellular system is designed with a directional antenna cellular configuration. A cluster pattern of 3 sector, size 7 is deployed. Compute the worst case signal to cochannel interference ratio S/I at the mobile receiver. If the S/I value for a practical system requires 6 dB higher than the theoretical value of 18 dB then comment on the results obtained. Assume the path loss exponent as 4. Solve for 6-sector N=7. Comment on the result. (Lab 5, Q 1)

```
cluster_size=[]
C_I_dB=[]

def sect3(clust):
    sect=3
    interfere_cell=2
    q=np.round((3*clust)**(0.5),1)
    print(f"Frequency reuse ratio q = {q}")
    C_I=np.floor((q**(-ple)+(q+0.7)**(-ple))**(-1))
    print(f"The C/I for given 3 sector antenna design is {C_I:.0f}")
    C_IdB=np.round(10*np.log10(C_I),1)
    print(f"The C/I for given 3 sector antenna design in dB is {C_IdB} higher than 18dB")
    cluster_size.append(clust)
    C_I_dB.append(C_IdB)

def sect6(clust):
    sect=6
    interfere_cell=1
    q=np.round((3*clust)**(0.5),1)
    print(f"Frequency reuse ratio q = {q}")
    C_I=np.floor(((q+0.7)**(-ple))**(-1))
    print(f"The C/I for given 6 sector antenna design is {C_I:.0f}")
    C_IdB=np.round(10*np.log10(C_I),1)
    print(f"The C/I for given 6 sector antenna design in dB is {C_IdB} higher than 18dB")
    cluster_size.append(clust)
    C_I_dB.append(C_IdB)


ple=4    #Path loss exponent 4 given
for i in range(1,5):
    print(f"Case {i}")
    sect=int(input("Enter the value of number of sectors : "))
    cluster=int(input("Enter the value of cluster size : "))
    if sect == 3 :
        sect3(cluster)
    elif (sect == 6) :
        sect6(cluster)
    print()
```

```
Case 2
Enter the value of number of sectors : 3
Enter the value of cluster size : 7
Frequency reuse ratio q = 4.6
The C/I for given 3 sector antenna design is 285
The C/I for given 3 sector antenna design in dB is 24.5 higher than 18dB
```

```
Case 4
Enter the value of number of sectors : 6
Enter the value of cluster size : 7
Frequency reuse ratio q = 4.6
The C/I for given 6 sector antenna design is 789
The C/I for given 6 sector antenna design in dB is 29.0 higher than 18dB
```

9. Compare the number of traffic channels per sector in the following two different cellular systems employing312 traffic channels for use: N=7 pattern with 3 sector configuration, N=4 pattern with 6 sector configuration (Lab 5, Q3)

```python
traff = 312
#System with reuse pattern K = 7 and 3 sector configuration
K = 7
sect = 3
print("**** For 1st System ****")
traf_chan_cell = np.round(traff/K)
print(f"Number of traffic channel per cell is {traf_chan_cell:.0f}")
traf_chan_sect = np.round(traf_chan_cell/sect)
print(f"Number of traffic channel per sector is {traf_chan_sect:.0f}")
print()

K = 4
sect = 6
print("**** For 2nd System ****")
traf_chan_cell = np.round(traff/K)
print(f"Number of traffic channel per cell is {traf_chan_cell:.0f}")
traf_chan_sect = np.round(traf_chan_cell/sect)
print(f"Number of traffic channel per sector is {traf_chan_sect:.0f}")
```

```
**** For 1st System ****
Number of traffic channel per cell is 45
Number of traffic channel per sector is 15

**** For 2nd System ****
Number of traffic channel per cell is 78
Number of traffic channel per sector is 13
```

10. Consider a transmitter which radiates a sinusoidal carrier frequency of 1850 MHz. For a vehicle moving 60mph, compute the received carrier frequency if the mobile is moving (a) directly towards the transmitter (b) directly away from the transmitter (c) in a direction which is perpendicular to the direction of arrival of the transmitted signal. (Lab 6)

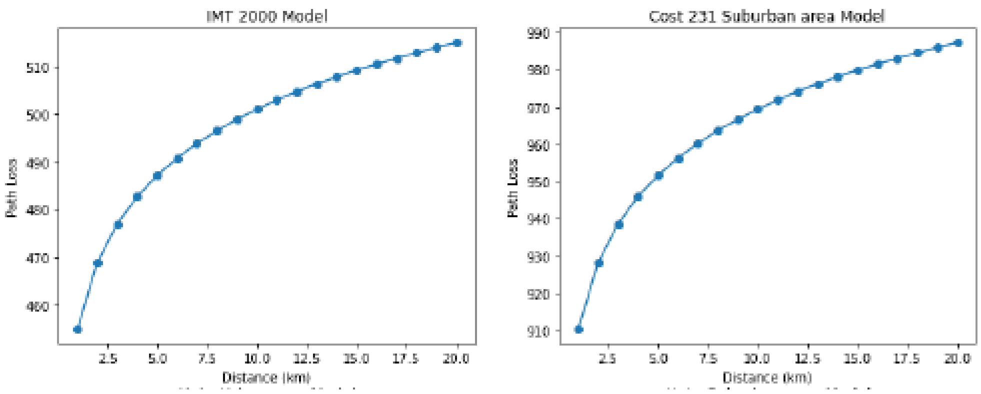
```python
Fc = 1850e6 #MHz
c = 3e8   #Speed of light
vehicle_spd = 60 #mph
wavelength = np.round((c/Fc),3)
print(f"The wavelength is {wavelength} m")
vehicle_spd = 1.61*vehicle_spd
vehicle_spd_ms = np.round((vehicle_spd * (5/18)),2)
print(f"The speed of the vehicles is {vehicle_spd_ms} m/s")
print()
print("The received carrier frequency (f) if the Vehicle (mobile) is moving..")
print()

# a). Directly towards the transmitter
Fd = np.round((vehicle_spd_ms/wavelength),2)
F = Fc + Fd
print(f"a. Directly towards the transmitter is : ")
print(f" f = fc + fd = {Fc:.0f} + {Fd} = {F} Hz\n")

# b). Directly away from the transmitter
Fd = np.round((vehicle_spd_ms/wavelength),2)
F = Fc - Fd
print(f"b. Directly away from the transmitter is : ")
print(f" f = fc - fd = {Fc:.0f} - {Fd} = {F} Hz\n")

# c). In a direction perpendicular to the direction of arrival of the transmitted signal
Fd = 0
F = Fc + Fd
print(f"c. In a direction perpendicular to the direction of arrival of the transmitted sign
print("Since the vehicle is moving perpendicular to the angle of arrival of the transmitted
print("Theta = 90 and cos(90) = 0")
print(f"Hence, f = fc + fd = {Fc:.0f} + {Fd} = {F:.0f} Hz\n")
```

```
The wavelength is 0.162 m
The speed of the vehicles is 26.83 m/s

The received carrier frequency (f) if the Vehicle (mobile) is moving..

a. Directly towards the transmitter is :
 f = fc + fd = 1850000000 + 165.62 = 1850000165.62 Hz

b. Directly away from the transmitter is :
 f = fc - fd = 1850000000 - 165.62 = 1849999834.38 Hz

c. In a direction perpendicular to the direction of arrival of the transmitt
ed signal is :
Since the vehicle is moving perpendicular to the angle of arrival of the tra
nsmitted signal,
Theta = 90 and cos(90) = 0
Hence, f = fc + fd = 1850000000 + 0 = 1850000000 Hz
```

11. Compute the path loss using the IMT 2000 and Cost 231 Suburban Area Models for the input parameters Centre frequency f = 1500-2000 MHz, f1 centre frequency f1 = 1800 MHz, Base station and Mobile station heights 50 km and 5 km respectively (Lab 7)

```
import numpy as np
import matplotlib.pyplot as plt
```

```
f=1500*(10**6)    #Centre freq in Hz
f1=1800*(10**6)    #Centre freq1 in Hz
hb=50              #base station height km
hm=5               #mobile station height km
a=3.2*((np.log(11.75*hm))**2) - 4.97
Gt=1
Gr=1
d=np.arange(1,21,1)
pl_imt= 32.4+20*np.log(d)+20*np.log(f)-10*np.log(Gt)-10*np.log(Gr)
pl_cost_suburb=46.3+(33.9*np.log(f1)-(13.82*np.log(hb)-(a*hm)+ \
(44.9-6.55*np.log(hb)*np.log(d))))
plt.figure(figsize=(20,10))
plt.subplot(2,3,1)
plt.plot(d,pl_imt,marker='o')
plt.xlabel("Distance (km)")
plt.ylabel("Path Loss")
plt.title("IMT 2000 Model")

plt.subplot(2,3,2)
plt.plot(d,pl_cost_suburb,marker='o')
plt.xlabel("Distance (km)")
plt.ylabel("Path Loss")
plt.title("Cost 231 Suburban area Model")
```



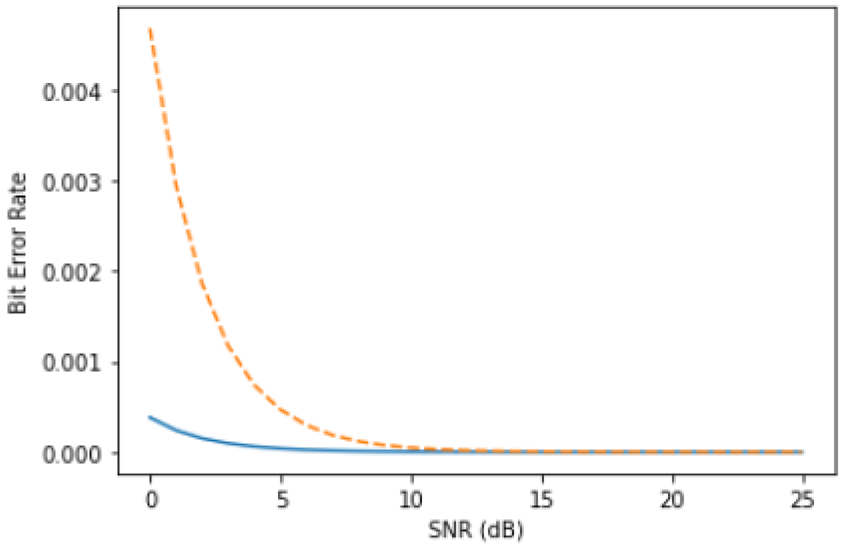
12. For the given parameters calculate BER and analyze the results N=256, 64 and L=3 taps, 4 taps (0<=SNRdB<=25) (Lab 8)

```
import numpy as np
from math import comb
import matplotlib.pyplot as plt
```

```
L = 3
N = 256
SNR_db = np.arange(0,26)
SNR = np.power(10,(SNR_db/10))
print(SNR)
a = comb(5,3)
b = 2*N*SNR
BER1 = np.power((a*(1/b)),2)
```

```
[  1.           1.25892541   1.58489319   1.99526231   2.51188643
   3.16227766   3.98107171   5.01187234   6.30957344   7.94328235
  10.          12.58925412  15.84893192  19.95262315  25.11886432
  31.6227766   39.81071706  50.11872336  63.09573445  79.43282347
 100.         125.89254118 158.48931925 199.5262315  251.18864315
 316.22776602]
```

```
L=4
a = comb(7,4)
b = 2*N*SNR
BER2 = np.power((a*(1/b)),2)
plt.plot(SNR_db,BER1)
plt.plot(SNR_db,BER2,"--")
plt.xlabel("SNR (dB)")
plt.ylabel('Bit Error Rate')
plt.show()
```



13. Generate 4 bit Walsh codes using Hadamard matrix (Lab 10)

```
import numpy as np
```

```
H=np.array([[0,0,0,0],[0,1,0,1],[0,0,1,1],[0,1,1,0]])
print("Hadamard Matrix :")
print(H)

b=[]
for i in range(len(H)):
    a=0
    for j in range(len(H)-1):
        if H[i][j]!=H[i][j+1]:
            a=a+1
    b.append(a)
c=np.sort(b)

W=[]
for i in range(len(H)):
    for j in range(len(H)):
        if c[i]==b[j]:
            W.append(H[j])
W1=np.ravel(W)
W2=np.reshape(W1,(len(H),len(H)))

print("Walsh Matrix (4Bit) : ")
print(W2)
```

```
Hadamard Matrix :
[[0 0 0 0]
 [0 1 0 1]
 [0 0 1 1]
 [0 1 1 0]]
Walsh Matrix (4Bit) :
[[0 0 0 0]
 [0 0 1 1]
 [0 1 1 0]
 [0 1 0 1]]
```

14. GSM uses the RPE-LTP speech coder in which the encoding is done on blocks of samples of 20-ms duration (260 bits of coder output). The most significant first 50 bits (Class Ia) are appended with 3 CRC bits, the next 132 bits (Class Ib) are appended by 4 tail bits and concatenated with the first error-protected bits. This block is then convolutionally encoded with a rate 1/2 FEC coder, and then concatenated with last 78 bits (Class II). Show that the achievable gross channel data rate is 22.8 kbps. (Lab 11)

```python
#  To compute number of encoded Class Ia bits
uncoded_Ia_bits = int(input("Enter number of uncoded Class Ia bits : "))
crc = int(input("Enter number of CRC bits : "))
coded_Ia_bits = uncoded_Ia_bits + crc
print(f"Number of encoded Class Ia bits are {coded_Ia_bits}")

#  To compute number of encoded Class Ib bits
uncoded_Ib_bits = int(input("Enter number of uncoded Class Ib bits : "))
tail = int(input("Enter number of tail bits : "))
coded_Ib_bits = uncoded_Ib_bits + tail
print(f"Number of encoded Class Ib bits are {coded_Ib_bits}")

#  To compute number of concatenation of encoded Class Ia + Class Ib bits
concat_bits = coded_Ia_bits + coded_Ib_bits
print(f"Number of concatenated bits are {concat_bits}")

#  To compute number of convolutionally encoded bits
conv_encoder_rate = float(input("Enter FEC coder rate of convolutional encoder : "))
conv_coded = (1 / conv_encoder_rate) * concat_bits
print(f"Number of convolutionally encoded bits are {conv_coded:.0f}")

#  To compute number of encoded bits to be transmitted
classII_bits = int(input("Enter number of Class II bits : "))
encode_bits_tx = conv_coded + classII_bits
print(f"Number of encoded bits to be transmitted are {encode_bits_tx:.0f}")

#  To determine achievable gross channel data rate
duration_tx = int(input("Enter duration of transmission (in ms): "))
gross_chan_bit_rate = encode_bits_tx / duration_tx
print(f"The gross channel bit rate is {gross_chan_bit_rate} kbps")
```

```
Enter number of uncoded Class Ia bits : 50
Enter number of CRC bits : 3
Number of encoded Class Ia bits are 53
Enter number of uncoded Class Ib bits : 132
Enter number of tail bits : 4
Number of encoded Class Ib bits are 136
Number of concatenated bits are 189
Enter FEC coder rate of convolutional encoder : 0.5
Number of convolutionally encoded bits are 378
Enter number of Class II bits : 78
Number of encoded bits to be transmitted are 456
Enter duration of transmission (in ms): 20
The gross channel bit rate is 22.8 kbps
```

15. Capacity of one carrier in a single-cell CDMA system : Given that the IS-95 CDMA digital cellular systems require 3 dB < Sr < 9 dB which employs QPSK modulation scheme and convolutional coding technique. The bandwidth of the channel is 1.25 MHz, and the transmission data rate is Rb = 9600 bps. Determine the capacity of a single IS-95 cell. (Lab 12)

```
#Maximum simultaneous users
Bc = 1250     # kHz
Rb = 9.6      # kbps
min_SR_db = 3
min_SR = round(10**(min_SR_db/10))
M_max = (Bc/Rb)*(1/min_SR)
print(f"The maximum no. of simultaneous users are {M_max:.0f}")

#Minimum simultaneous users
max_SR_db = 9
max_SR = round(10**(max_SR_db/10))
M_min = (Bc/Rb)*(1/max_SR)
print(f"The minimum no. of simultaneous users are {M_min:.0f}")
print(f"Hence, a single cell IS-95 CDMA digital cellular system can support \
from {M_min:.0f} users to {M_max:.0f} users")
```

```
The maximum no. of simultaneous users are 65
The minimum no. of simultaneous users are 16
Hence, a single cell IS-95 CDMA digital cellular system can support from 16 users to 65
users
```