**BER of BPSK with PYTHON**

(1) The best way to develop Python codes is to use Google Colaboratory

https://colab.research.google.com

A general tutorial for Python is available at

https://www.w3schools.com/python/

(2) Our goal is to convert the given MATLAB code into Python. However, we need to keep certain things in mind.

a) One must be very careful about the spaces used before a new line, especially while using nested for loops.
b) Python handles the arrays differently, the array index starts at 0 (just like C or C++) unlike 1 in MATLAB.
c) We don't need to put a semicolon after every command.
d) The comment starts with a # (% in MATLAB).

(3) Some functions in 'math' library handle only scalar values, for array operation, we may have to import 'special'. For example, the function erfc() in 'math' library can only handle scalar values. So, in order to evaluate the theoretical BER, one can use:

```
from scipy import special
import numpy as np
SNRdB    = np.arange(-10,12,2)          #Signal to Noise Ratio (in dB)
SNR      = np.power(10,SNRdB/10)        #Convert SNR from dB to normal
Arg_Erfc = np.sqrt(SNR)                 #Calculate argument of erfc
BER_th   = (1/2)*special.erfc(Arg_Erfc) #Calculate theoretical BER
```

Note that SNR range is -10 to 10 dB in steps of 2 dB [last value, 12, is not included]

(4) Initialization of an array can be tricky in Python. For example, the code in MATLAB, `zeros(1,length(SNR))`, in Python is something like:

```
BER_sim  = [0]*len(SNR)                 #Initialize simulation BER array
```

(5) For baseband modulation, AWGN generation and detection, the following code snippet might be useful:

```
#Generate binary data
data  = np.random.randint(2,size=num_bit)
#Baseband BPSK modulation
S     = 2*data-1
#Generate AWGN
mu    = 0
sigma = np.sqrt(N0[count]/2)
N     = sigma * np.random.randn(num_bit) + mu
#Received Signal
Y     = S + N
#Decision device taking hard decision and deciding error
for k in range(num_bit):
```

```
        if (Y[k]>0 and data[k]==0) or (Y[k]<0 and data[k]==1): Error = Error + 1
```

(6) With the help of the above code, make two nested for loops like the code in MATLAB and run the simulation. Plot the results, the following code snippet may be used.

```
import matplotlib.pyplot as plt          #for plotting
#Plot BER
plt.plot(SNRdB,BER th,'r')
plt.plot(SNRdB,BER sim,'ko')
plt.legend(["Analytical","Simulation"])
plt.yscale('log')
plt.grid(True)
plt.ylabel('BER')
plt.xlabel('SNR [dB]')
plt.show()
```

(7) Once an output like the following is obtained, put the Python code and the output figure in a single PDF file and submit.