



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
HAROKOPIO UNIVERSITY

ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ ΣΤΟ ΜΑΘΗΜΑ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΑ ΣΥΣΤΗΜΑΤΑ
2022-2023
ΔΡΑΜΠΑΛΟΣ ΕΥΑΓΓΕΛΟΣ
21823

Μέρος 1ο: η συνάρτηση Q

A)

Αρχικά μετατρέπονται οι συναρτήσεις σε κώδικα python.

```
def qfunction(x):  
    return 0.5*special.erfc(x/np.sqrt(2))
```

```
def A1(x):  
    return np.exp(-(x**2)/2)
```

```
def A2(x):  
    return (0.25*np.exp(-x**2)+0.25*np.exp(-(x**2)/2))
```

```
def A3(x):  
    return (0.0833*np.exp(-(x**2)/2)+0.25*np.exp(-(2*(x**2))/3))
```

περιορίζουμε τις τιμές για $2 \leq x \leq 7$ με 100 σημεία

```
x = np.linspace(2.0, 7.0, 100)
```

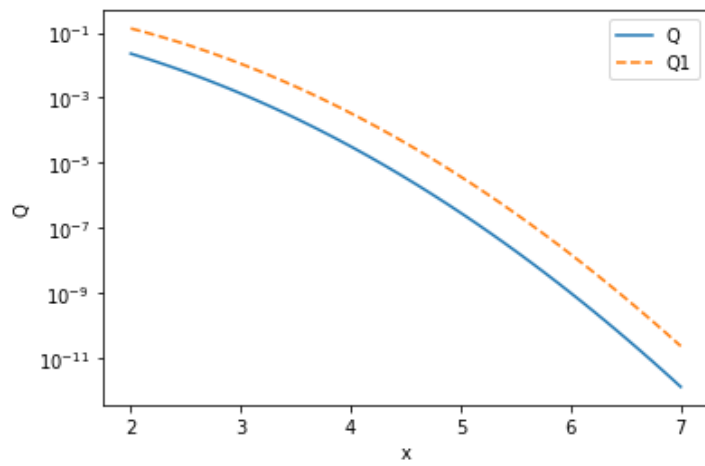
κλήση των συναρτήσεων

```
Q = qfunction(x)  
Q1= A1(x)  
Q2=A2(x)  
Q3=A3(x)
```

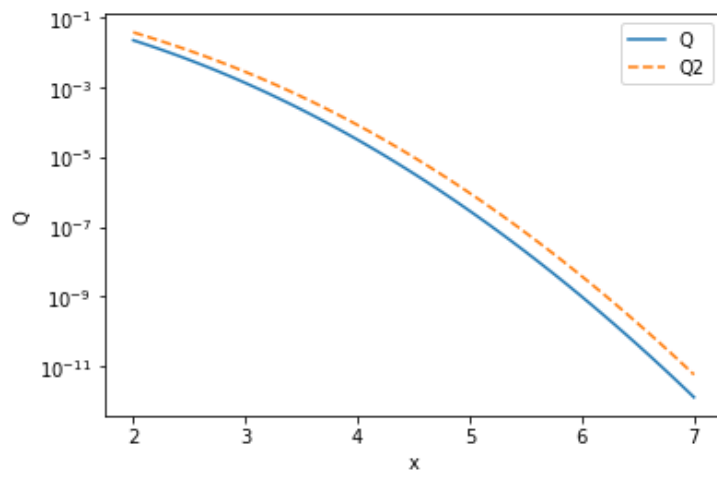
Δημιουργία γραφημάτων

```
plt.close('all')  
plt.figure(1)  
plt.semilogy(x, Q, label='Q')  
plt.semilogy(x, Q1, linestyle='dashed', label='Q1')  
plt.xlabel('x')  
plt.ylabel('Q')  
plt.legend()  
plt.show()  
plt.figure(2)  
plt.semilogy(x, Q, label='Q')  
plt.semilogy(x, Q2, linestyle='dashed', label='Q2')  
plt.xlabel('x')  
plt.ylabel('Q')  
plt.legend()  
plt.show()  
plt.figure(3)  
plt.semilogy(x, Q, label='Q')  
plt.semilogy(x, Q3, linestyle='dashed', label='Q3')  
plt.xlabel('x')  
plt.ylabel('Q')  
plt.legend()  
plt.show()
```

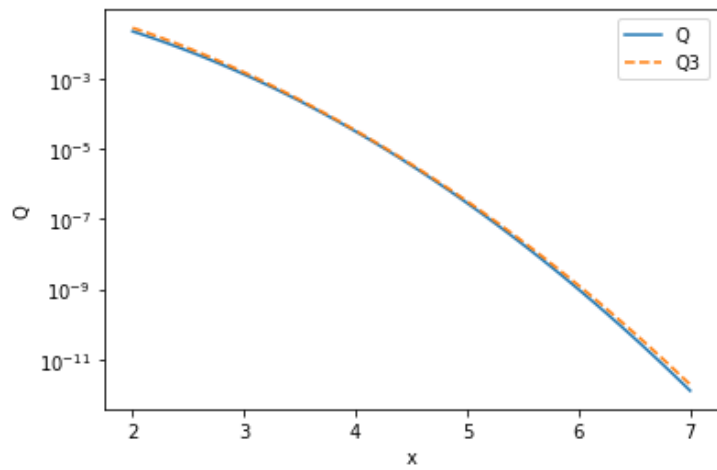
A1: $Q(x)-Q1(x)$



A2: $Q(x)-Q2(x)$



A3: $Q(x)-Q3(x)$



B)

t για αναθεση στο διαστημα 2 εως 7 με 1000 σημεια

C1,C2,C3 είναι αριθμητες του κλασματος

g παρονομαστης

γινεται χρηση της **trapz** για το καθε αντιστοιχο **Qi-Q**

```
t= np.linspace(2.0, 7.0, 1000)
c1=np.absolute(A1(t)-qfunction(t))
c2=np.absolute(A2(t)-qfunction(t))
c3=np.absolute(A3(t)-qfunction(t))
g=np.absolute(qfunction(t))
I1=np.trapz(c1/g, t)
I2=np.trapz(c2/g, t)
I3=np.trapz(c3/g, t)
```

1. Για **Q1(x)-Q(x)** είναι **54.15924314118071**
2. Για **Q2(x)-Q(x)** είναι **9.88665615127752**
3. Για **Q3(x)-Q(x)** είναι **0.9944553926065828**

Ποια είναι η καλύτερη περίπτωση;

Απάντηση: είναι η **3** αφού βλέπουμε οτι με διαφορά είναι ο μικρότερος αριθμός άρα η καλύτερη προσέγγιση.

Μέρος 2ο: Κυματομορφή PAM

εισαγωγή του ονοματος στο val

μετατροπή του του val σε bits στην μεταβλητή res

```
val = input("Enter your value: ")  
res = ''.join(format(ord(i), '08b') for i in val)
```

Θετουμε λιστα με τα bits

```
bitlist = [res[i:i+1] for i in range(0, len(res), 1)]
```

δημιουργία χαρτη για το pam 2 με κωδικοποίηση gray

και θετω το Ts με βάση την εκφωνηση

```
mdictionary2 = cl.pam_gray_forward_map(2)  
TS = 1*1e-9
```

Θελω την λιστα bitlist να γίνει np array στο bits

στο symbolarray με βάση το mapping που δινουμε φτιαχνει τα συμβολα PAM και μετα φτιαχεται η κυματομορφή συμφωνα με το Ts που εχουμε ορισει

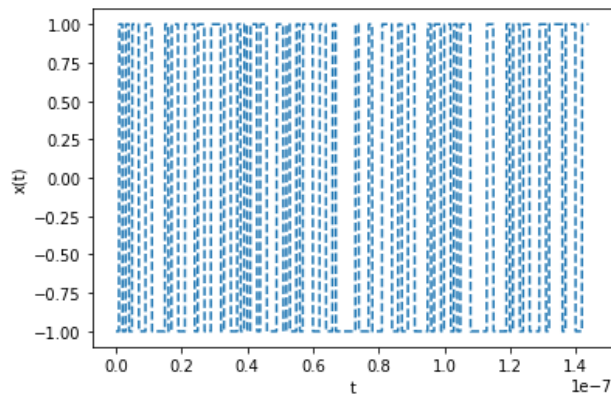
το cl.plot_signal() το σχεδιαζει

```
bits = np.array(bitlist)  
symbolarray = cl.bits_to_symbols(bits, mdictionary2)  
t, x = cl.pam_waveform2(symbolarray, TS)  
  
cl.plot_signal(t, x, plot_type='--', figure_no=1)
```

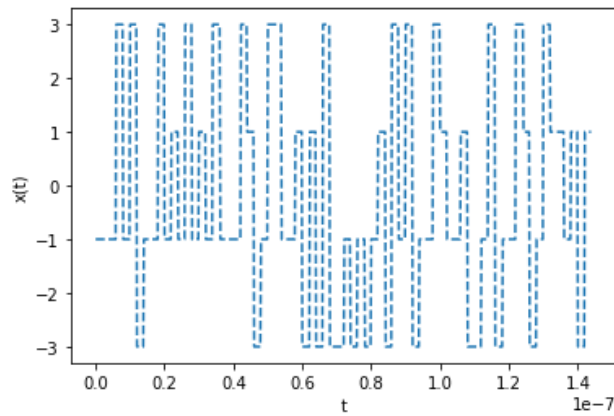
ο Κωδικας επαναλαμβανεται με μικρες διαφορες για το 4-PAM ,16-PAM(το 8-PAM δεν το έχω κάνει)

Input: Vaggelis Drampalos

1.M=2

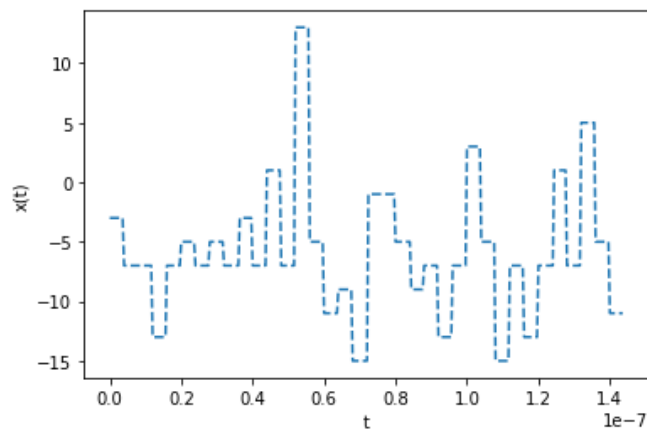


2.M=4



3.Για M=8 δεν το έχω κάνει

4.M=16



ΑΝΑΦΟΡΕΣ ΓΙΑ ΤΟ ΜΕΡΟΣ 2:

“απαρτίζει” το string που γίνεται input από τον χρήστη:

<https://pythonexamples.org/python-split-string-into-specific-length-chunks/>

Για να παίρνει το πρόγραμμα Input:

<https://www.geeksforgeeks.org/taking-input-in-python/>

string to binary:

<https://www.geeksforgeeks.org/python-convert-string-to-binary/>

Χρησιμοποιήθηκε κώδικας από το μάθημα:

<https://github.com/thomaskamalakis/telecomsystems>