Étude d'une fonction de transfert

⚠ Le nombre complexe i est ici noté j.

◇ On utilisera :

```
from math import *
from cmath import phase
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import fsolve
```

Le nombre complexe $z = a + jb$ s'écrit avec python **`z=a+1j*b`**, son module est obtenu avec **`abs(z)`** et **`phase(z)`** est un argument de z.

On considère un filtre dont la fonction de transfert est :

$$H(\omega) = \frac{1/4}{1 + jQ\left(\frac{\omega}{\omega_0} - \frac{\omega_0}{\omega}\right)}$$

avec $Q = 10$ et $f_0 = \dfrac{\omega_0}{2\pi} = 1000$ Hz.

1 Quelques tracés

Question 1 : Définir les constantes de l'énoncé ainsi que la fonction H (à valeurs complexes).

Question 2 : Représenter graphiquement $|H|$ et $\arg(H)$ pour $\omega \in [\omega_0/10, 10\omega_0]$.

Question 3 : Effectuer les mêmes représentations graphiques mais avec une échelle logarithmique pour ω.

Question 4 : On considère la fonction G définir par $G(\omega) = 20\log(|H(\omega)|)$. Définir la fonction G et calculer $G(\omega_0)$.

Question 5 : Représenter les fonctions G et $\arg(H)$ dans la même fenêtre, en échelle logarithmique pour ω.

Question 6 : Sur la représentation graphique précédente, tracer la droite horizontale correspondant à l'ordonnée $G(\omega_0) - 3$.

2 Recherche des fréquences de coupure à -3 dB

Question 7 : À l'aide de la représensation graphique, estimer numériquement les valeurs ω_1 et ω_2 solutions de l'équation $G(\omega) = G(\omega_0) - 3$.

Question 8 : Résoudre numériquement l'équation $G(\omega) = G(\omega_0) - 3$ en utilisant **`fsolve`**. Déterminer ω_1 et ω_2 puis les fréquences f_1 et f_2 associées.

Question 9 : Vérifier numériquement que $f_0 = \sqrt{f_1 f_2}$ (moyenne géométrique) et $Q = \dfrac{f_0}{\Delta f}$.

3 Comportement à haute et basse fréquences

Question 10 : Déterminer un équivalent de $H(\omega)$ puis de $|H(\omega)|$ lorsque $\omega \to +\infty$.

Question 11 : En utilisant l'équivalent précédent, déterminer un développement asymptotique de $G(\omega)$ de la forme :

$$G(\omega) = a\log(\omega) + b + \underset{\omega\to+\infty}{\mathrm{o}}(1)$$

Représenter graphiquement la fonction $\omega \mapsto a\log\omega + b$ sur le même dessin que G.

Question 12 : Faire de même pour $\omega \to 0$.

Question 13 : Comparer avec les mêmes représentations graphiques sans échelle logarithmique.

Corrections

Q. 1 :

```
Q=10.0
f0=1000.0
w0=2*pi*f0
```

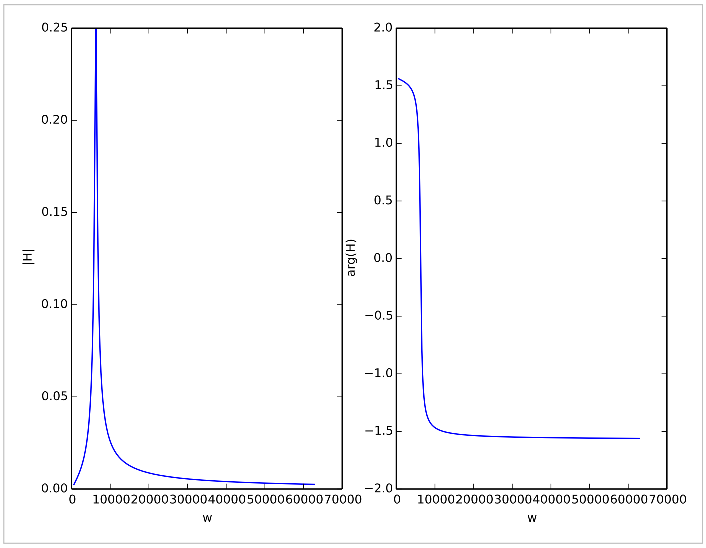
```
def H(w):
    return 0.25/(1+1j*Q*(w/w0-w0/w))
```

Remarque importante : cette fonction **H** pourra s'appliquer aussi bien à des nombres (w de type *float*) qu'à des tableaux (w de type *array*).

Q. 2 : On définit **w**, intervalle allant de **w0/10** à **10*w0** et comportant 1000 valeurs régulièrements espacées. On réalise ensuite les représentations graphiques de la manière habituelle :

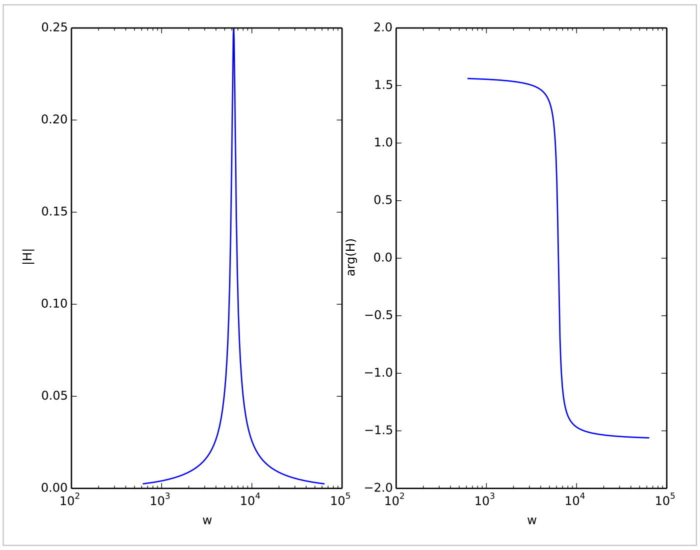
- pour $|H(\omega)|$, on considère **H(w)** (tableau de nombres complexes) et on lui applique la fonction **np.abs** ;
- on ne peut pas procéder de la même manière pour l'argument car la fonction **phase** ne s'applique pas à des tableaux. On utilise par conséquent la construction **[phase(H(wi)) for wi in w]**.

```
w=np.linspace(w0/10,w0*10,1000)
plt.clf()
plt.subplot(1,2,1)
plt.plot(w,np.abs(H(w)))
plt.xlabel("w")
plt.ylabel("|H|")
plt.subplot(1,2,2)
plt.plot(w,[phase(H(wi)) for wi in w])
plt.xlabel("w")
plt.ylabel("arg(H)")
```



Q. 3 : Pour réaliser des représentations graphiques avec une échelle logarithmique en abscisses, il suffit d'utiliser **semilogx** au lieu de **plot** :

```
plt.clf()
plt.subplot(1,2,1)
plt.semilogx(w,np.abs(H(w)))
plt.xlabel("w")
plt.ylabel("|H|")
plt.subplot(1,2,2)
plt.semilogx(w,[phase(H(wi)) for wi in w])
plt.xlabel("w")
plt.ylabel("arg(H)")
```



Q. 4 :

```
def G(w):
    return 20*np.log10(np.abs(H(w)))
```

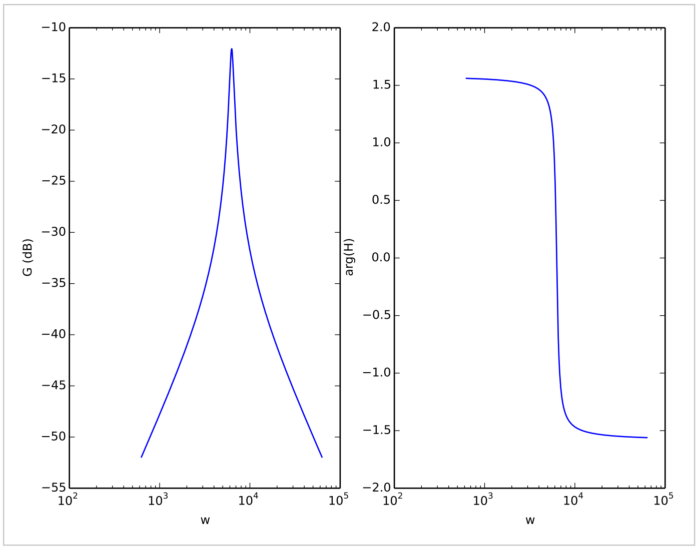
En utilisant **np.log10** et **np.abs**, la fonction **G** pourra s'appliquer aussi bien à des nombres qu'à des tableaux.

```
print G(w0)
```

```
-12.0411998266
```

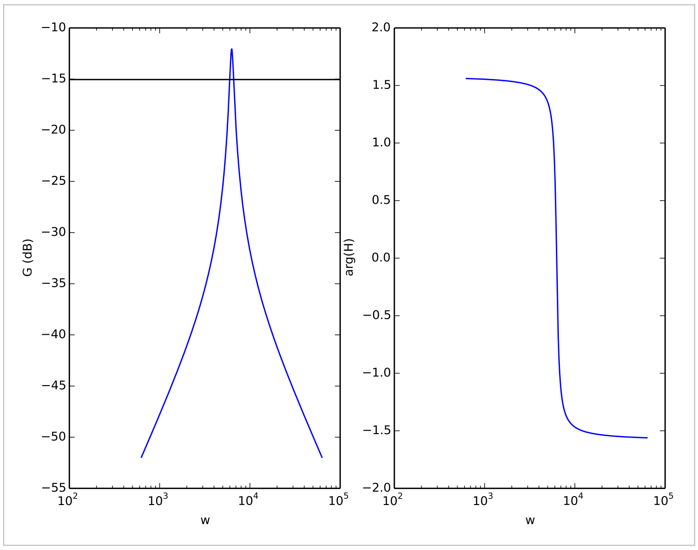
Q. 5 :

```
plt.clf()
plt.subplot(1,2,1)
plt.semilogx(w,G(w))
plt.xlabel("w")
plt.ylabel("G (dB)")
plt.subplot(1,2,2)
plt.semilogx(w,[phase(H(wi)) for wi in w])
plt.xlabel("w")
plt.ylabel("arg(H)")
```



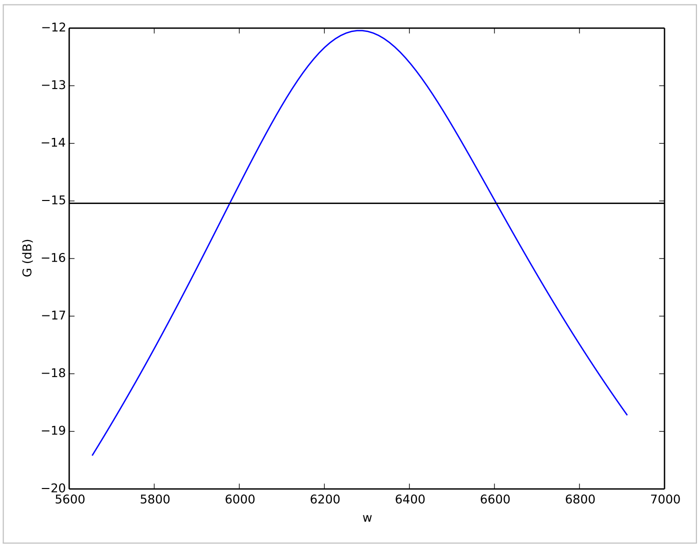
Q. 6 :

```
plt.clf()
plt.subplot(1,2,1)
plt.semilogx(w,G(w))
plt.axhline(y=G(w0)-3,color='k')
plt.xlabel("w")
plt.ylabel("G (dB)")
plt.subplot(1,2,2)
plt.semilogx(w,[phase(H(wi)) for wi in w])
plt.xlabel("w")
plt.ylabel("arg(H)")
```



Q. 7 : Pas très clair sur le dessin quand même. On peut faire une représentation plus adaptée :

```
plt.clf()
wmini=np.linspace(2*pi*900,2*pi*1100,100)
plt.plot(wmini,G(wmini))
plt.axhline(y=G(w0)-3,color='k')
plt.xlabel("w")
plt.ylabel("G (dB)")
```



Q. 8 : On va utiliser la fonction **fsolve**. Il faut lui donner l'équation à résoudre sous la forme $f(x) = 0$, on définit donc :

```
def f(w):
    return G(w)-G(w0)+3
```

Il faut également fournir à **fsolve** une valeur de départ. L'équation considérée possède deux solutions, une à gauche de ω_0 et une à droite. On va donc partir de deux valeurs, l'une inférieure à ω_0 (par exemple $\omega_0/10$) et l'autre supérieure (par exemple $10\omega_0$) :

```
print fsolve(f,w0/10)
print fsolve(f,w0*10)
```

`[ 5977.58303482] [-5977.58303482]` Visiblement, il y a un problème pour la fréquence la plus haute. On va donc partir de plus haut :

```
print fsolve(f,w0/10)
print fsolve(f,w0*100)
```

`[ 5977.58303482] [ 6604.41141083]` C'est cohérent, on définit donc :

```
w1=fsolve(f,w0/10)[0]
w2=fsolve(f,w0*100)[0]
f1=w1/2/pi
f2=w2/2/pi
print "f1=%f Hz, f2=%f Hz" % (f1,f2)
```

`f1=951.361888 Hz, f2=1051.124722 Hz`

Q. 9 :

```
print "f0=%f Hz, sqrt(f1*f2)=%f Hz" % (f0,sqrt(f1*f2))
```

```
f0=1000.000000 Hz, sqrt(f1*f2)=1000.000000 Hz
```

```
print "Q=%f, f0/Delta f=%f" %(Q,f0/(f2-f1))
```

```
Q=10.000000, f0/Delta f=10.023773
```

Q. 10 :

- Le physicien dit : lorsque ω est grand,

$$H(\omega) \simeq \frac{1/4}{jQ\frac{\omega}{\omega_0}}$$

et ainsi :

$$|H(\omega)| \simeq \frac{\omega_0}{4Q\omega}$$

- Le mathématicien dit (sensiblement la même chose en fait) :

$$1 + jQ\frac{\omega}{\omega_0} - jQ\frac{\omega_0}{\omega} \underset{\omega\to+\infty}{\sim} jQ\frac{\omega}{\omega 0}$$

donc en prenant l'inverse :

$$H(\omega) \underset{\omega\to+\infty}{\sim} \frac{1/4}{jQ\frac{\omega}{\omega_0}} = \frac{\omega_0}{4jQ\omega}$$

puis en prenant le module :

$$|H(\omega)| \underset{\omega\to+\infty}{\sim} \left|\frac{\omega_0}{4jQ\omega}\right| = \frac{\omega_0}{4Q\omega}$$

En effet, si $u(x) \underset{x\to x_0}{\sim} v(x)$, alors $\dfrac{u(x)}{v(x)} \xrightarrow[x\to x_0]{} 1$, donc $\left|\dfrac{u(x)}{v(x)}\right| \xrightarrow[x\to x_0]{} 1$ et $|u(x)| \underset{x\to x_0}{\sim} |v(x)|$.

Q. 11 :

- Le physicien dit : lorsque ω est grand,

$$|H(\omega)| \simeq \frac{\omega_0}{4Q\omega}$$

donc en appliquant le logarithme :

$$G(\omega) \simeq 20\log\frac{\omega_0}{4Q\omega} = 20\log\frac{\omega_0}{4Q} - 20\log\omega$$

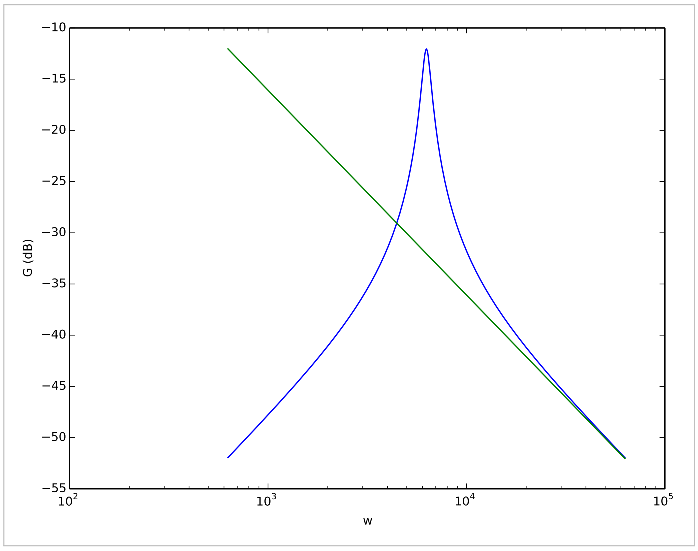
- Le mathématicien dira plutôt :

$$G(\omega) = 20\log|H(\omega)| = \underbrace{20\log\frac{|H(\omega)|}{\frac{\omega_0}{4Q\omega}}}_{\xrightarrow[\omega\to+\infty]{}0} + 20\log\frac{\omega_0}{4Q\omega} = 20\log\frac{\omega_0}{4Q} - 20\log\omega + \underset{\omega\to+\infty}{\mathrm{o}}(1)$$

car, en général, $u(x) \underset{x\to x_0}{\sim} v(x)$ *n'implique pas* $\log(u(x)) \underset{x\to x_0}{\sim} \log(v(x))$.

Représentation graphique :

```
plt.clf()
plt.semilogx(w,G(w))
plt.semilogx(w,20*log10(w0/Q/4)-20*np.log10(w),'g')
plt.xlabel("w")
plt.ylabel("G (dB)")
```

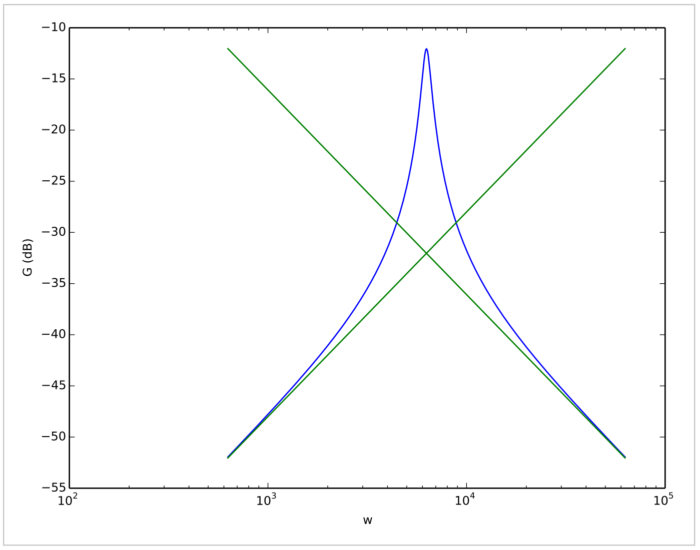


Q. 12 : Avec les mêmes techniques, on obtient :

$$|H(\omega)| \underset{\omega \to 0}{\sim} \frac{\omega}{4Q\omega_0}$$
$$G(\omega) = 20 \log \frac{1}{4Q\omega_0} + 20 \log \omega + \underset{\omega \to 0}{o}(1)$$

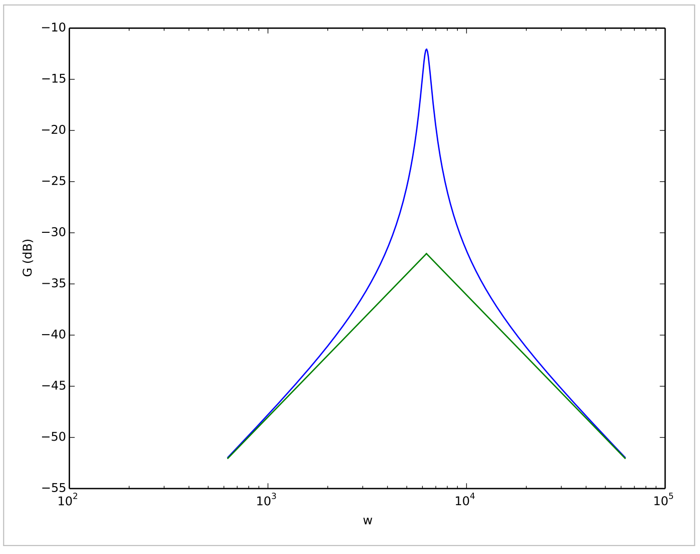
Représentation graphique :

```
plt.clf()
plt.semilogx(w,G(w))
plt.semilogx(w,20*log10(w0/Q/4)-20*np.log10(w),'g')
plt.semilogx(w,20*log10(1/w0/Q/4)+20*np.log10(w),'g')
plt.xlabel("w")
plt.ylabel("G (dB)")
```



On peut arrêter les asymptotes à ω_0 :

```
plt.clf()
plt.semilogx(w,G(w))
w0moins=np.linspace(w0/10,w0,100)
w0plus=np.linspace(w0,w0*10,100)
plt.semilogx(w0plus,20*log10(w0/Q/4)-20*np.log10(w0plus),'g')
plt.semilogx(w0moins,20*log10(1/w0/Q/4)+20*np.log10(w0moins),'g')
plt.xlabel("w")
plt.ylabel("G (dB)")
```



Q. 13 :

```
plt.clf()
plt.plot(w,G(w))
w0moins=np.linspace(w0/10,w0,100)
w0plus=np.linspace(w0,w0*10,100)
plt.plot(w0plus,20*log10(w0/Q/4)-20*np.log10(w0plus),'g')
plt.plot(w0moins,20*log10(1/w0/Q/4)+20*np.log10(w0moins),'g')
plt.xlabel("w")
plt.ylabel("G (dB)")
```