calc_mesure_faible

March 28, 2023

[ ]:
```python
#sympy pour effectuer les calcules
import sympy as smp
from IPython.display import display, Math
smp.init_session()
```

```
IPython console for SymPy 1.11.1 (Python 3.11.2-64-bit) (ground types: python)

These commands were executed:
>>> from sympy import *
>>> x, y, z, t = symbols('x y z t')
>>> k, m, n = symbols('k m n', integer=True)
>>> f, g, h = symbols('f g h', cls=Function)
>>> init_printing()

Documentation can be found at https://docs.sympy.org/1.11.1/
```

[ ]:
```python
a = smp.symbols('a', real=False, complex = True)
t = smp.symbols('t', real=True)
o = smp.symbols(r'\sigma', real=True, positive = True)
b = smp.symbols('b', real=False, complex = True)
d = smp.symbols(r'\delta', real=True)
```

[ ]:
```python
#Fonction gaussien
A = (smp.sqrt(1/((smp.sqrt(2*smp.pi))*o)))*smp.exp(-(t**2)/(4*o**2))
display(Math(r'A(t) = '+smp.latex(A)))
```

$$A(t) = \frac{2^{\frac{3}{4}} e^{- \frac{t^{2}}{4 \sigma^{2}}}}{2 \sqrt[4]{\pi} \sqrt{\sigma}}$$

[ ]:
```python
#état polarisé horizontal
h = a*A
display(Math('H(t) = ' +smp.latex(h)))
```

$$H(t) = \frac{2^{\frac{3}{4}} a e^{- \frac{t^{2}}{4 \sigma^{2}}}}{2 \sqrt[4]{\pi} \sqrt{\sigma}}$$

[ ]:
```
#état polarisé vertical
v = b*A
display(Math('V(t) = ' +latex(v)))
```

$$V(t) = \frac{2^{\frac{3}{4}}be^{-\frac{t^2}{4\sigma^2}}}{2\sqrt[4]{\pi}\sqrt{\sigma}}$$

[ ]:
```
#intiallement
psi = h + v
display(Math(r'\psi(t) = ' +smp.latex(psi)))
```

$$\psi(t) = \frac{2^{\frac{3}{4}}ae^{-\frac{t^2}{4\sigma^2}}}{2\sqrt[4]{\pi}\sqrt{\sigma}} + \frac{2^{\frac{3}{4}}be^{-\frac{t^2}{4\sigma^2}}}{2\sqrt[4]{\pi}\sqrt{\sigma}}$$

[ ]:
```
#decalage (mesure faible) sur H
A_f = (smp.sqrt(1/((smp.sqrt(2*smp.pi))*o)))*smp.exp(-((t-d)**2)/(4*o**2))
h_f = A_f*a
psi_f = (h_f + v)
display(Math(r'\psi(t)_f = ' +smp.latex(psi_f)))
```

$$\psi(t)_f = \frac{2^{\frac{3}{4}}ae^{-\frac{(-\delta+t)^2}{4\sigma^2}}}{2\sqrt[4]{\pi}\sqrt{\sigma}} + \frac{2^{\frac{3}{4}}be^{-\frac{t^2}{4\sigma^2}}}{2\sqrt[4]{\pi}\sqrt{\sigma}}$$

[ ]:
```
diag_f = (1/smp.sqrt(2))*(h_f + v)
display(Math(r'D(t)_f = ' +smp.latex(diag_f)))
```

$$D(t)_f = \frac{\sqrt{2}\cdot\left(\frac{2^{\frac{3}{4}}ae^{-\frac{(-\delta+t)^2}{4\sigma^2}}}{2\sqrt[4]{\pi}\sqrt{\sigma}} + \frac{2^{\frac{3}{4}}be^{-\frac{t^2}{4\sigma^2}}}{2\sqrt[4]{\pi}\sqrt{\sigma}}\right)}{2}$$

[ ]:
```
psi_a = (1/smp.sqrt(2))*(h_f)
psi_b = (1/smp.sqrt(2))*(v)
display(Math(r'\psi(t)_a = ' +smp.latex(psi_a)))
display(Math(r'\psi(t)_b = ' +smp.latex(psi_b)))
```

$$\psi(t)_a = \frac{\sqrt[4]{2}ae^{-\frac{(-\delta+t)^2}{4\sigma^2}}}{2\sqrt[4]{\pi}\sqrt{\sigma}}$$

$$\psi(t)_b = \frac{\sqrt[4]{2}be^{-\frac{t^2}{4\sigma^2}}}{2\sqrt[4]{\pi}\sqrt{\sigma}}$$

[ ]:
```
psi_1 = psi_a*smp.conjugate(psi_a)
psi_2 = psi_a*smp.conjugate(psi_b)
psi_3 = smp.conjugate(psi_a)*psi_b
psi_4 = smp.conjugate(psi_b)*psi_b
display(Math(r'\psi(t)_1 = ' +smp.latex(psi_1)))
display(Math(r'\psi(t)_2 = ' +smp.latex(psi_2)))
display(Math(r'\psi(t)_3 = ' +smp.latex(psi_3)))
```

```
display(Math(r'\psi(t)_4 = ' +smp.latex(psi_4)))
```

$$\psi(t)_1 = \frac{\sqrt{2}ae^{-\frac{(-\delta+t)^2}{2\sigma^2}}\overline{a}}{4\sqrt{\pi}\sigma}$$

$$\psi(t)_2 = \frac{\sqrt{2}ae^{-\frac{t^2}{4\sigma^2}}e^{-\frac{(-\delta+t)^2}{4\sigma^2}}\overline{b}}{4\sqrt{\pi}\sigma}$$

$$\psi(t)_3 = \frac{\sqrt{2}be^{-\frac{t^2}{4\sigma^2}}e^{-\frac{(-\delta+t)^2}{4\sigma^2}}\overline{a}}{4\sqrt{\pi}\sigma}$$

$$\psi(t)_4 = \frac{\sqrt{2}be^{-\frac{t^2}{2\sigma^2}}\overline{b}}{4\sqrt{\pi}\sigma}$$

[ ]:
```
psi_1_t = smp.integrate(psi_1*t, (t, -smp.oo, smp.oo))
psi_2_t = smp.integrate(psi_2*t, (t, -smp.oo, smp.oo))
psi_3_t = smp.integrate(psi_3*t, (t, -smp.oo, smp.oo))
psi_4_t = smp.integrate(psi_4*t, (t, -smp.oo, smp.oo))

display(Math(r'\psi(t)_1 = ' +smp.latex(psi_1_t)))
display(Math(r'\psi(t)_2 = ' +smp.latex(psi_2_t)))
display(Math(r'\psi(t)_3 = ' +smp.latex(psi_3_t)))
display(Math(r'\psi(t)_4 = ' +smp.latex(psi_4_t)))
```

$$\psi(t)_1 = \frac{\delta a\overline{a}}{2}$$

$$\psi(t)_2 = \frac{\delta a\left(2-\operatorname{erfc}\left(\frac{\sqrt{2}\delta}{4\sigma}\right)\right)e^{-\frac{\delta^2}{8\sigma^2}}\overline{b}}{4} + \frac{\delta a\left(-2\sqrt{\pi}\left(2-\operatorname{erfc}\left(\frac{\sqrt{2}\delta}{4\sigma}\right)\right)-\frac{4\sqrt{2}\sigma e^{-\frac{\delta^2}{8\sigma^2}}}{\delta}\right)e^{-\frac{\delta^2}{8\sigma^2}}\overline{b}}{16\sqrt{\pi}} + \frac{\delta a\left(-2\sqrt{\pi}\operatorname{erfc}\left(\frac{\sqrt{2}\delta}{4\sigma}\right)+\frac{4\sqrt{2}\sigma e^{-\frac{\delta^2}{8\sigma^2}}}{\delta}\right)e^{-\frac{\delta^2}{8\sigma^2}}\overline{b}}{16\sqrt{\pi}} + \frac{\delta ae^{-\frac{\delta^2}{8\sigma^2}}\overline{b}\operatorname{erfc}\left(\frac{\sqrt{2}\delta}{4\sigma}\right)}{4}$$

$$\psi(t)_3 = \frac{\delta b\left(2-\operatorname{erfc}\left(\frac{\sqrt{2}\delta}{4\sigma}\right)\right)e^{-\frac{\delta^2}{8\sigma^2}}\overline{a}}{4} + \frac{\delta b\left(-2\sqrt{\pi}\left(2-\operatorname{erfc}\left(\frac{\sqrt{2}\delta}{4\sigma}\right)\right)-\frac{4\sqrt{2}\sigma e^{-\frac{\delta^2}{8\sigma^2}}}{\delta}\right)e^{-\frac{\delta^2}{8\sigma^2}}\overline{a}}{16\sqrt{\pi}} + \frac{\delta b\left(-2\sqrt{\pi}\operatorname{erfc}\left(\frac{\sqrt{2}\delta}{4\sigma}\right)+\frac{4\sqrt{2}\sigma e^{-\frac{\delta^2}{8\sigma^2}}}{\delta}\right)e^{-\frac{\delta^2}{8\sigma^2}}\overline{a}}{16\sqrt{\pi}} + \frac{\delta be^{-\frac{\delta^2}{8\sigma^2}}\overline{a}\operatorname{erfc}\left(\frac{\sqrt{2}\delta}{4\sigma}\right)}{4}$$

$$\psi(t)_4 = 0$$

[ ]:
```
t_moy = smp.simplify(psi_1_t + psi_2_t + psi_3_t + psi_4_t)
display(Math("<T>"+ smp.latex(t_moy)))

t_moy_limit = t_moy.subs(smp.exp((-d**2)/(8*o**2)), 1)
display(Math(r'\lim_{e^{\frac{-\delta^{2}}{8\sigma^{2}}} \to 1} <T>= ' +smp.
↪latex(t_moy_limit)))
```

$$< T > \frac{\delta \left(2ae^{\frac{\delta^2}{8\sigma^2}}\overline{a} + a\overline{b} + b\overline{a}\right)e^{-\frac{\delta^2}{8\sigma^2}}}{4}$$

$$\lim_{e^{\frac{-\delta^2}{8\sigma^2}} \to 1} < T >= \frac{\delta\left(2a\overline{a} + a\overline{b} + b\overline{a}\right)}{4}$$

[ ]:
```
#continue avec ce système pour trouver la moyenne de la fréquence
w = smp.symbols(' ', real=True)

d_ff = smp.fourier_transform(diag_f, t, w)
display(Math(r'\psi( )_{f_t} = ' +smp.latex(d_ff)))
```

$$\psi(\omega)_{f_t} = \sqrt[4]{2}\sqrt[4]{\pi}\sqrt{\sigma}\left(a + be^{2i\pi\delta\omega}\right)e^{-2\pi\omega(i\delta+2\pi\sigma^2\omega)}$$

[ ]:
```
#transformation de Fourier
f1 = smp.fourier_transform(psi_1, t, w)
f2 = smp.fourier_transform(psi_2, t, w)
f3 = smp.fourier_transform(psi_3, t, w)
f4 = smp.fourier_transform(psi_4, t, w)
```

[ ]:
```
#combinaisons des postsélections de D
f_11 = f1*smp.conjugate(f1)
f_22 = f2*smp.conjugate(f2)
f_33 = f3*smp.conjugate(f3)
f_44 = f4*smp.conjugate(f4)
f_12 = f1*smp.conjugate(f2)
f_13 = f1*smp.conjugate(f3)
f_14 = f1*smp.conjugate(f4)
f_21 = f2*smp.conjugate(f1)
f_23 = f2*smp.conjugate(f3)
f_24 = f2*smp.conjugate(f4)
f_31 = f3*smp.conjugate(f1)
f_32 = f3*smp.conjugate(f2)
f_34 = f3*smp.conjugate(f4)
f_41 = f4*smp.conjugate(f1)
f_42 = f4*smp.conjugate(f2)
f_43 = f4*smp.conjugate(f3)
```

[ ]:
```
#integre chaque partie
f_11_w = smp.integrate(f_11*w, (w, -smp.oo, smp.oo))
f_22_w = smp.integrate(f_22*w, (w, -smp.oo, smp.oo))
f_33_w = smp.integrate(f_33*w, (w, -smp.oo, smp.oo))
f_44_w = smp.integrate(f_44*w, (w, -smp.oo, smp.oo))
f_12_w = smp.integrate(f_12*w, (w, -smp.oo, smp.oo))
f_13_w = smp.integrate(f_13*w, (w, -smp.oo, smp.oo))
f_14_w = smp.integrate(f_14*w, (w, -smp.oo, smp.oo))
f_21_w = smp.integrate(f_21*w, (w, -smp.oo, smp.oo))
```

```
f_23_w = smp.integrate(f_23*w, (w, -smp.oo, smp.oo))
f_24_w = smp.integrate(f_24*w, (w, -smp.oo, smp.oo))
f_31_w = smp.integrate(f_31*w, (w, -smp.oo, smp.oo))
f_32_w = smp.integrate(f_32*w, (w, -smp.oo, smp.oo))
f_34_w = smp.integrate(f_34*w, (w, -smp.oo, smp.oo))
f_41_w = smp.integrate(f_41*w, (w, -smp.oo, smp.oo))
f_42_w = smp.integrate(f_42*w, (w, -smp.oo, smp.oo))
f_43_w = smp.integrate(f_43*w, (w, -smp.oo, smp.oo))
```

[ ]:

```
#trouve la fréquence moyenne
w_moy = smp.simplify(f_11_w + f_22_w + f_33_w + f_44_w + f_12_w + f_13_w +␣
 ↪f_14_w + f_21_w + f_23_w + f_24_w + f_31_w + f_32_w + f_34_w + f_41_w +␣
 ↪f_42_w + f_43_w)
display(Math(r'<\hat{\Omega}> = ' +smp.latex(w_moy)))

#prend limit de exp((-d**2)/(8*o**2) -> 1
w_moy_limit = w_moy.subs(smp.exp((-d**2)/(8*o**2)), 1)
display(Math(r'\lim_{e^{\frac{-\delta^{2}}{8\sigma^{2}}} \to 1} <\hat{\Omega}>␣
 ↪= ' +smp.latex(w_moy_limit)))
```

$$<\hat{\Omega}>=0$$

$$\lim_{e^{\frac{-\delta^{2}}{8\sigma^{2}}} \to 1} <\hat{\Omega}>=0$$

[ ]:

```
#Trouver la valeur faible
W_d_h = (1/d)*(t_moy_limit - 4*smp.I*(o**2)*w_moy_limit)
display(Math(r'<\hat{S}>_{W} = ' +smp.latex(W_d_h)))
#La partie réel de la valeur faible de D est
display(Math(r'<\hat{t}> = ' + smp.latex(smp.simplify(smp.re(W_d_h)))))
display(Math(r'<\hat{\omega}> = ' + smp.latex(smp.simplify(smp.im(W_d_h)))))
```

$$<\hat{S}>_{W}=\frac{a\overline{a}}{2}+\frac{a\overline{b}}{4}+\frac{b\overline{a}}{4}$$

$$<\hat{t}>=\frac{\operatorname{re}\left(a\overline{b}\right)}{4}+\frac{\operatorname{re}\left(b\overline{a}\right)}{4}+\frac{|a|^{2}}{2}$$

$$<\hat{\omega}>=\frac{\operatorname{im}\left(a\overline{b}\right)}{4}+\frac{\operatorname{im}\left(b\overline{a}\right)}{4}$$