

Paquet d'onde et dispersion

1. Introduction

Ce document montre comment faire la synthèse de paquets d'ondes par la somme de fonctions d'ondes sinusoïdales. On verra l'effet de la relation de dispersion du milieu sur l'évolution temporelle d'un paquet d'onde. La simulation s'applique aux ondes électromagnétiques, mais aussi aux ondes d'amplitude de probabilité en mécanique quantique (ondes de De Broglie).

2. Principe

2.a. Ondes périodiques

Considérons tout d'abord une onde plane sinusoïdale (en notation complexe) :

$$u(x, t) = A \exp[i(kx - \omega t)] \quad (1)$$

On appelle *relation de dispersion* la fonction $k(\omega)$ donnant le nombre d'onde en fonction de la pulsation. La relation de dispersion dépend du milieu dans lequel l'onde se propage. On utilisera les relations de dispersion suivantes :

▷ Vide

$$k(\omega) = \frac{\omega}{c} \quad (2)$$

▷ Milieu à coupure (plasma)

$$k(\omega) = \frac{1}{c} \sqrt{\omega^2 - \omega_e^2} \quad (3)$$

▷ Milieu transparent (indice optique)

$$k(\omega) = \frac{\omega}{c} (1 + b\omega^2) \quad (4)$$

▷ Onde d'une particule non relativiste libre (onde de De Broglie)

$$k(\omega) = \sqrt{\frac{2m\omega}{\hbar}} \quad (5)$$

Soit une onde générée à partir d'une source périodique, de période T . La pulsation fondamentale est définie par :

$$\omega_1 = \frac{2\pi}{T} \quad (6)$$

La série de Fourier permet d'écrire l'onde comme une somme d'ondes sinusoïdales :

$$u(x, t) = \sum_{n=1}^P A_n \exp[i(k(n\omega_1)x - n\omega_1 t)] \quad (7)$$

L'harmonique de rang n est une onde sinusoïdale de pulsation $n\omega_1$, dont le nombre d'onde est $k(n\omega_1)$. L'amplitude A_n de cette harmonique est en général un nombre complexe.

La vitesse de phase est la vitesse de déplacement d'une surface de phase constante :

$$V_\phi = \frac{\omega}{k(\omega)} \quad (8)$$

À l'exception du cas d'une onde électromagnétique dans le vide, la vitesse de phase dépend de la pulsation. Cela signifie que les différentes harmoniques ne se propagent pas exactement à la même vitesse. En conséquence, la forme de l'onde est en général modifiée au cours du temps. Ce phénomène est appelé *dispersion de l'onde*.

2.b. Paquet d'onde

En principe, un paquet d'onde est une onde localisée dans le temps (et dans l'espace), c'est-à-dire une impulsion plus ou moins brève. Il ne s'agit pas d'une onde périodique. On peut néanmoins construire une onde périodique constituée d'une succession périodique de paquets d'onde, de la manière suivante.

Pour simplifier la description, on pose $T = 1$, ce qui revient à identifier l'indice n de l'harmonique à la fréquence. On considère un spectre constitué d'une fréquence centrale f (un nombre entier) assez élevée (au moins 100), et de P raies de part et d'autres de cette fréquence. Le spectre est donc constitué des $2P + 1$ raies $f - P, f - P + 1, \dots, f + P$. Toutes les autres harmoniques sont nulles. On suppose de plus que P est petit devant f . Le nombre de raies est impair si P est entier, mais on peut choisir P demi-entier pour avoir un nombre de raies pair.

Comme l'indice n dans la série de Fourier reste relativement proche de f , on fait le développement limité suivant (à l'ordre 1) :

$$k(n\omega_1) = k(f\omega_1) + (n - f)\omega_1 k'(f\omega_1) \quad (9)$$

où l'on a introduit la fonction $k'(\omega)$, dérivée de la relation de dispersion.

On pose $q = n - f$, un indice qui varie de $-P$ à $+P$ dans la somme. La phase de l'harmonique de rang n s'écrit alors :

$$\phi(x, t) = k(f\omega_1)x - f\omega_1 t + q\omega_1 k'(f\omega_1)x - q\omega_1 t \quad (10)$$

Après factorisation, on obtient donc l'expression suivante de l'onde :

$$u(x, t) = \exp[i(k(f\omega_1)x - f\omega_1 t)] \sum_{q=-P}^P \exp[iq\omega_1(k'(f\omega_1)x - t)] \quad (11)$$

L'exponentielle en facteur est une onde sinusoïdale se propageant à la vitesse de phase de la fréquence f . La somme constitue une enveloppe, dont la vitesse de propagation est :

$$V_g = \frac{1}{k'(f\omega_1)} = \frac{d\omega}{dk}(f\omega_1) \quad (12)$$

Cette vitesse est la *vitesse de groupe*. C'est la vitesse de déplacement des paquets d'onde. Dans un milieu dispersif, la vitesse de groupe est différente de la vitesse de phase.

Nous avons défini le paquet d'onde à partir de son spectre temporel : les fréquences temporelles ω sont multiples d'une fréquence fondamentale et les fréquences spatiales k sont calculées avec la relation de dispersion $k(\omega)$. On peut aussi définir le paquet d'onde à partir de son spectre spatial : les fréquences spatiales k sont alors multiples d'une fréquence fondamentale et les fréquences temporelles sont calculées avec la relation $\omega(k)$.

Dans le vide, les deux descriptions sont équivalentes. En radio-transmission, on utilise plutôt un spectre en fréquence correspondant au spectre du signal émis.

3. Calcul numérique

Pour le calcul numérique, on pose $T = 1$ et on simplifie les relations de dispersion en posant $c = 1$. Par ailleurs, on pose :

$$a = \frac{2\pi f}{k(2\pi f)} \quad (13)$$

Le nombre d'onde sera multiplié par a et la vitesse de groupe sera divisée par a . Cela permet d'avoir toujours $k = 2\pi f$ pour la fréquence centrale du paquet d'onde, indépendamment de la relation de dispersion choisie.

Les différentes fonctions sont dans une classe, dont voici le constructeur :

[onde.py](#)

```
import numpy
import math
from scipy.signal import get_window

class Onde:
    def __init__(self, suivi=True, dispersion='vide', coupure=0.0, coeff=1.0):
        """
            : param suivi : suivi du paquet d'onde a la vitesse de groupe
            : param dispersion : relation de dispersion
            : param coupure : frequence de coupure pour la relation de dispersion 'co
            : param coeff : coefficient pour la relation de dispersion 'optique'
        """
        self.freq = [1.0, 3.0, 5.0]
        self.amp = [1.0, 0.5, 0.2]
        self.phase = [0.0, 0.0, 0.0]
        self.nf = len(self.freq)
        if suivi:
            self.suivi = 1.0
        else:
            self.suivi = 0.0
        self.coupure = coupure
        self.coeff = coeff
        self.wc2 = (2*math.pi*coupure)**2
        self.a = 1.0

        if dispersion=='vide':
            self.k = self.k_vide
            self.vg = self.vg_vide

        elif dispersion=='coupure':
            self.k = self.k_coupure
            self.vg = self.vg_coupure
```

```
elif dispersion=='optique':
    self.k = self.k_optique
    self.vg = self.vg_optique

elif dispersion=='debroglie':
    self.k = self.k_debroglie
    self.vg = self.vg_debroglie
```

Voici les différentes relations de dispersion :

```
def k_vide(self,w):
    return w
def vg_vide(self,w):
    return 1.0

def k_coupure(self,w):
    return math.sqrt(w*w-self.wc2)
def vg_coupure(self,w):
    return math.sqrt(1-self.wc2/(w*w))

def k_optique(self,w):
    return w*(1.0+self.coeff*w*w)
def vg_optique(self,w):
    return 1.0/(1.0+3*self.coeff*w*w)

def k_debroglie(self,w):
    return math.sqrt(w)
def vg_debroglie(self,w):
    return 2.0*math.sqrt(w)
```

La fonction suivante définit un paquet d'onde. Le fenêtrage permet de choisir la forme du spectre autour de la fréquence centrale. On peut utiliser 'boxcar', 'hamming', ou les autres fenêtrages disponibles dans la fonction [scipy.signal.get window](#).

```
def paquet(self,f,P,window="hamming"):
    """
        Creation d'un paquet d'onde
        : param f : frequence centrale
        : param P : 2P+1 = nombre de frequences
        : param window : fenetrage
    """
    M = int(2*P+1)
    self.a = 2*math.pi*f/self.k(2*math.pi*f)
    self.freq = numpy.zeros(M)
    self.amp = numpy.zeros(M)
    self.phase = numpy.zeros(M)
```

```

self.nf = M
self.vgroupe = self.vg(2*math.pi*f)/self.a
for n in range(M):
    self.freq[n] = f-P+n
    self.amp[n] = 1.0
    self.phase[n] = 0.0
self.amp = get_window(window,M)

```

La fonction suivante effectue l'échantillonnage de la partie réelle de l'onde à l'instant t , sur un intervalle de positions donné. Si le paramètre `suivi` est égal à 1, le paquet d'onde est suivi à la vitesse de groupe de sa fréquence centrale.

```

def echantillons(self,xmin,xmax,t,N):
    """
        calcul de N echantillons de l'onde reelle sur l'intervalle [xmin,xmax] a
    """
    x = numpy.linspace(xmin,xmax,N)
    y = numpy.zeros(x.size)
    for i in range(self.nf):
        w = 2*math.pi*self.freq[i]
        k = self.k(w)*self.a
        phi = k*x+(k*self.vgroupe*self.suivi-w)*t+self.phase[i]
        y += self.amp[i]*numpy.cos(phi)
    return (x,y)

```

La fonction suivante effectue la même chose mais renvoie le module au carré de la fonction d'onde complexe. Pour une onde électromagnétique, cela correspond à l'intensité moyenne de l'onde. Pour une onde de De Broglie, cela correspond à la densité de probabilité associée à l'onde.

```

def echantillons_proba(self,xmin,xmax,t,N):
    """
        calcul de N echantillons de la densite de probabilite sur l'intervalle [x
    """
    x = numpy.linspace(xmin,xmax,N)
    yr = numpy.zeros(x.size)
    yi = numpy.zeros(x.size)
    for i in range(self.nf):
        w = 2*math.pi*self.freq[i]
        k = self.k(w)*self.a
        phi = k*x+(k*self.vgroupe*self.suivi-w)*t+self.phase[i]
        yr += self.amp[i]*numpy.cos(phi)
        yi += self.amp[i]*numpy.sin(phi)
    return (x,numpy.absolute(yr+1j*yi))

```

4. Exemples

```
import numpy
import math
from matplotlib.pyplot import *
import matplotlib.animation as animation
from onde import Onde
```

4.a. Dispersion avec coupure

On choisit une fréquence de coupure de 50 :

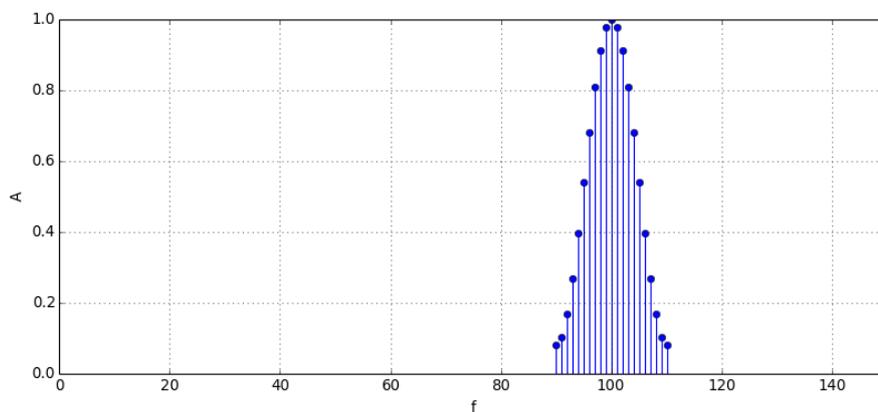
```
onde = Onde(suivi=True,dispersion='coupure',coupure=50)
```

On définit un paquet d'onde avec une fréquence centrale de 100 et 10 fréquences de part et d'autre, avec un fenêtrage de Hamming :

```
onde.paquet(100,10>window='hamming')
```

Voyons tout d'abord le spectre :

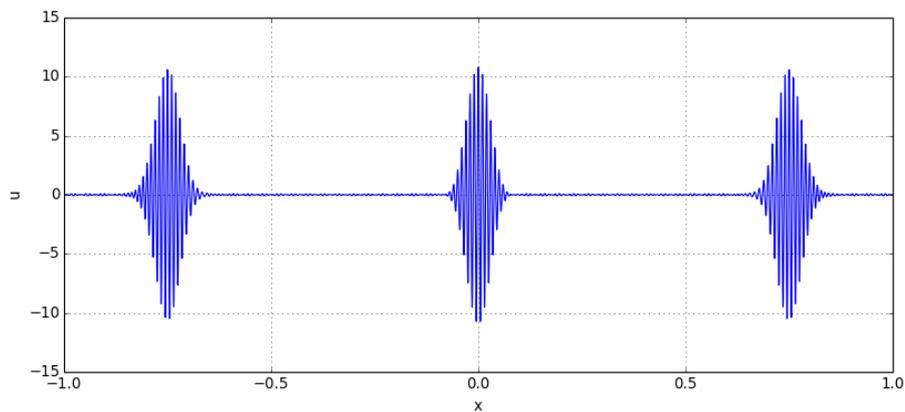
```
figure(figsize=(12,5))
stem(onde.freq,onde.amp)
xlabel("f")
ylabel("A")
grid()
axis([0,150,0,1.0])
```



Voici l'onde à l'instant initial :

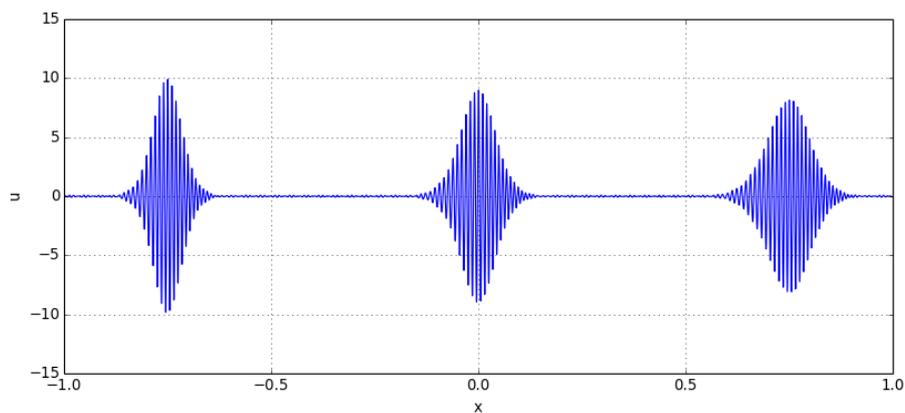
```
temps = 0.0
N = 5000
xmin = -1.0
xmax = 1.0
```

```
(x,y) = onde.echantillons(xmin,xmax,temps,N)
figure(figsize=(12,5))
plot(x,y)
xlabel("x")
ylabel("u")
axis([xmin,xmax,-15,15])
grid()
```



Les paquets se déplacent à la vitesse de groupe. Comme le suivi à la vitesse de groupe est activé, on peut voir la forme des paquets un instant plus tard :

```
temps=3.0
(x,y) = onde.echantillons(xmin,xmax,temps,N)
figure(figsize=(12,5))
plot(x,y)
xlabel("x")
ylabel("u")
axis([xmin,xmax,-15,15])
grid()
```



On observe un étalement des paquets.

Il est possible d'obtenir une animation montrant l'évolution dans le temps des paquets. Voici comment procéder :

```
temps = 0.0
dt = 0.05
N = 5000
xmin = -2.0
xmax = 2.0

(x,y) = onde.echantillons(xmin,xmax,temps,N)

fig, ax = subplots()
line, = ax.plot(x,y)
ax.grid()

def animate(i):
    global temps,xmin,xmax,N
    temps += dt
    (x,y) = onde.echantillons(xmin,xmax,temps,N)
    line.set_xdata(x)
    line.set_ydata(y)
    return line,

ani = animation.FuncAnimation(fig,animate,1000,interval=40)

show()
```

La fonction `animate` est appelée 1000 fois, toutes les 40 millisecondes. À chaque appel, on incrémente le temps de `dt`. En modifiant cette valeur, on peut changer la vitesse de l'animation.

L'animation permet de bien voir la différence entre la vitesse de groupe et la vitesse de phase, et permet de suivre l'étalement progressif des paquets d'onde.

4.b. Onde de De Broglie

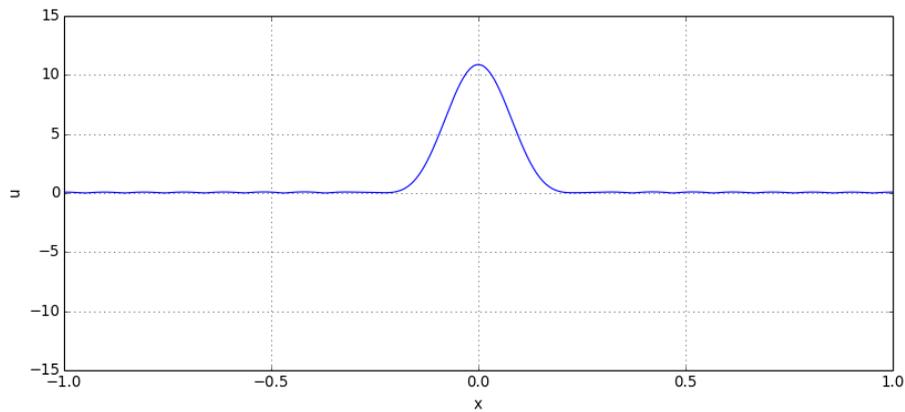
L'onde de De Broglie associée à une particule libre définit une amplitude de probabilité. Son module au carré donne la probabilité de trouver la particule à l'abscisse x .

```
onde = Onde(suivi=True,dispersion='debroglie')
onde.paquet(100,10>window='hamming')
```

Voici la densité de probabilité à l'instant initial :

```
temps = 0.0
N = 5000
xmin = -1.0
xmax = 1
(x,y) = onde.echantillons_proba(xmin,xmax,temps,N)
figure(figsize=(12,5))
plot(x,y)
```

```
xlabel("x")
ylabel("u")
axis([xmin,xmax,-15,15])
grid()
```



Voyons la densité de probabilité un instant plus tard :

```
temps=3.0
(x,y) = onde.echantillons_proba(xmin,xmax,temps,N)
figure(figsize=(12,5))
plot(x,y)
xlabel("x")
ylabel("u")
axis([xmin,xmax,-15,15])
grid()
```

