Capacité numérique : à l'aide d'un langage de programmation, simuler la propagation d'un paquet d'ondes dans un milieu dispersif et visualiser le phénomène d'étalement.

Considérons une O.P.P scalaire quelconque $s(x,t)\in\mathbb{R}$ se propageant selon les x croissants, *sans atténuation*, dans un milieu linéaire dispersif. Utilisons la décomposition fréquentielle de la perturbation $s(x=0,t)$, supposée être une fonction paire de t pour simplifier :

$s(x=0,t)=F(t)=\dfrac{1}{\pi}\displaystyle\int_0^{+\infty}\tilde{F}(\omega)\cos(\omega t)\mathrm{d}\omega$, où $\omega\mapsto\tilde{F}(\omega)$ est la transformée de Fourier de $t\mapsto F(t)$, c'est-à-dire qu'elle caractérise le spectre de ce signal.

La composante $\tilde{F}(\omega)\cos(\omega t)$ de pulsation ω à l'instant t et en $x=0$ se propage à la vitesse de phase $v_\varphi(\omega)=\omega/k(\omega)$: elle s'écrit $\tilde{F}(\omega)\cos\left[\omega t-k(\omega)x\right]$ en x quelconque.

Le milieu étant linéaire, il suffit, pour déterminer le signal $s(x,t)$, de sommer toutes ses composantes sinusoïdales en x à l'instant t : $\boldsymbol{s(x,t)=\dfrac{1}{\pi}\displaystyle\int_0^{+\infty}\tilde{F}(\omega)\cos\left[\omega t-k(\omega)x\right]\mathrm{d}\omega}$. Cette formule permet de connaître la forme de l'onde à tout instant, si on connaît son spectre en $x=0$ et la relation de dispersion $k(\omega)$.

Pour un spectre continu quelconque $\omega\mapsto\tilde{F}(\omega)$, le calcul de $s(x,t)$ nécessite un calcul approché d'intégrale pour chaque valeur de x et chaque valeur de t, ce qui implique un grand nombre de boucles.

Méthode numérique

On peut approcher l'intégrale par la *méthode des trapèzes* pour raccourcir les temps de calcul. L'intégrale de f entre a et $b\geq a$ est alors approximée par la somme des aires des trapèzes obtenus en découpant $[a,b]$ en N intervalles dont les bornes sont :

$x_0=a,\ldots,x_i=a+i\times h,\ldots,x_N=b$, où le pas d'intégration est $h=\dfrac{b-a}{N}$.

L'aire du $i^{\text{ème}}$ trapèze est $\dfrac{1}{2}(x_i-x_{i-1})\times\left[f(x_{i-1})+f(x_i)\right]=\dfrac{h}{2}\times\left[f(x_{i-1})+f(x_i)\right]$ d'où :

$$\int_a^b f(x)\mathrm{d}x=\frac{h}{2}\sum_{i=1}^{N}\left[f(x_{i-1})+f(x_i)\right]=\frac{h}{2}\left[f(a)+f(b)+2\sum_{i=1}^{N-1}f(x_i)\right].$$

Q.1) On suppose que l'on dispose de deux fonctions :

— **k**`(omega)` qui renvoie la valeur de la pulsation spatiale k correspondant à la valeur entrée pour la pulsation `omega` (relation de dispersion $\omega\mapsto k(\omega)$).

— **f**`(omega)` qui renvoie la valeur de la transformée de Fourier $\omega\mapsto\tilde{F}(\omega)$ de F correspondant à la valeur entrée pour la pulsation `omega`.

Écrire une fonction **`trapezes`**`(f,k,omega_min,omega_max,N,x,t)` qui renvoie la valeur de $s(x,t)$ obtenue grâce à l'intégrale $s(x,t) = \frac{1}{\pi}\int_{\omega_{min}}^{\omega_{max}} \tilde{F}(\omega)\cos[\omega t - k(\omega)x]\mathrm{d}\omega$, approchée grâce à la méthode des trapèzes.

— `x` et `t` sont les valeurs de l'abscisse x et du temps t.

— `omega_min` et `omega_max` sont les valeurs extrêmes ω_{min} et ω_{max} choisies pour la pulsation, et dépendent du spectre du signal temporel.

— `N` est le nombre d'intervalles de pulsations entre ω_{min} et ω_{max}.

Paquet d'onde rectangulaire étroit

On s'intéresse à un paquet d'onde rectangulaire étroit, c'est-à-dire que $\tilde{F}(\omega) = 0$ sauf pour $\omega \in [\omega_{min} = \omega_0 - \Delta\omega/2, \omega_{max} = \omega_0 + \Delta\omega/2]$, avec $\Delta\omega << \omega_0$, où $\tilde{F}(\omega) = F_0 = Cte$. On peut alors linéariser la relation de dispersion, qui devient $k(\omega) = k(\omega_0) + (\omega - \omega_0)\frac{\mathrm{d}k}{\mathrm{d}\omega}(\omega_0) = \frac{\omega_0}{v_\varphi} + \frac{\omega - \omega_0}{v_g}$. v_φ et v_g sont respectivement la vitesse de phase pour la pulsation ω_0.

On sait dans ce cas calculer littéralement le signal $s(x,t)$:

$s(x,t) = \frac{F_0 \Delta\omega}{\pi}\mathrm{sinc}\left[\frac{\Delta\omega}{2}\left(t - \frac{x}{v_g}\right)\right]\cos[\omega_0 t - k(\omega_0)x]$. La valeur numérique de F_0 est fixée à 1.

Q.2) Comparer, en les traçant sur le même graphe, le signal $s(x=0,t)$, obtenu par intégration numérique avec la méthode des trapèzes, avec sa valeur théorique. On pourra prendre $\omega_0 = 1$, $\Delta\omega = \omega_0/20$ et représenter $t \mapsto F(t)$ sur l'intervalle $[-4\pi/\Delta\omega, 4\pi/\Delta\omega]$ pour 1000 intervalles temporels.

Attention : sous numpy, la fonction sinc est en fait la fonction $t \mapsto \mathrm{sinc}(\pi t)$.

Faire varier la valeur de N et commenter.

Q.3) Réaliser une animation de l'onde en représentant successivement `NT = 1000` courbes pour $t \in [-4\pi/\Delta\omega, 4\pi/\Delta\omega]$.

On fait varier l'abscisse x entre $-4\pi/\Delta k$ et $+4\pi/\Delta k$, où $\Delta k = k(\omega_{max}) - k(\omega_{min})$. Justifier ce choix.

On fixe la vitesse de groupe `vg = 1`, et on prend successivement pour la vitesse de phase : `v_phi = 1`, puis `0.6` puis `1.5`. Commenter le phénomène observé.

Paquet d'onde gaussien

On s'intéresse maintenant à un paquet d'onde gaussien, centré sur la pulsation ω_0, et d'écart-type ω_1 : $\tilde{F}(\omega) = F_0 e^{-\frac{(\omega-\omega_0)^2}{2\omega_1^2}}$. Il est plus réaliste que le paquet d'onde rectangulaire précédent.

La fonction $\tilde{F}$ décroissant rapidement quand on s'écarte de ω_0, on peut considérer que la bande spectrale autour ω_0 a pour largeur $\Delta\omega = 4\omega_1$.

Cette fois-ci, le paquet d'onde n'est pas nécessairement étroit, et on prend la relation de dispersion suivante : $k(\omega) = \dfrac{\sqrt{\omega^2 - {\omega_c}^2}}{c}$, où ω_c est la pulsation de coupure.

Pour appliquer la méthode des trapèzes précédente, on prendra $\omega_{min} = \omega_c$ et $\omega_{max} = \omega_0 + 3\omega_1$.

Q.4) Réaliser une animation de l'onde en représentant successivement `NT = 1000` courbes pour $t \in [0\ ,\ 30 \times 2\pi / \Delta\omega]$.

On fait varier l'abscisse x entre $-x_{max}/10$ et $x_{max} = 60 \times 2\pi / \Delta k$, où :

$\Delta k = k(\omega_0 + 2\omega_1) - k(\omega_0 - 2\omega_1)$. Justifier ce choix.

On fixe `c = 1`, et on prend pour pulsation centrale `omega0 = 6`, pour pulsation de coupure `omegac = 2`, et pour écart-type `omega1 = 1.5`, puis `2`, puis `0.5`. Commenter le phénomène observé.

Faire une animation sous matplotlib

On utilise la méthode `FuncAnimation` de la bibliothèque animation.

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

X = np.linspace(0,5,500) # on observe sur 5 longueurs d'onde

def f(x,t): # O.P.P.H dans le sens des x croissants
    return np.cos(2*np.pi*(t-x))

def g(x,t): # O.P.P.H dans le sens des x décroissants
    return np.cos(2*np.pi*(c*t+x))

def s(x,t): # onde stationnaire résultante
    return f(x,t)+g(x,t)

# I) Initialisation : on crée une figure avec des limites pour les
axes.

ma_figure = plt.figure()
plt.xlim(0,5)
plt.ylim(-2,2)

# II) Création de la fonction qui sera appelée à chaque nouvelle image

t_max = 10
N = 200
dt = t_max/(N-1)

def mon_animation(i):
    t = i*dt
    courbe1, = plt.plot(X,f(X,t),color = 'black',lw = 2)
    courbe2, = plt.plot(X,g(X,t),color = 'red',lw = 2)
    courbe3, = plt.plot(X,s(X,t),color = 'green',lw = 2)
```

```
    return courbe1,courbe2,courbe3

# III) Génération de l'animation

mon_animation = FuncAnimation(fig = ma_figure, func = mon_animation,
frames = N, interval = 100, blit = True, repeat = True, repeat_delay =
1000)

plt.show()
```

La syntaxe dans le return de la fonction `mon_animation` peut surprendre. Il faut savoir que l'objet renvoyé par `plot` est un itérable de longueur 1. La syntaxe `courbe, = plt.plot( )` permet d'affecter à la variable courbe le contenu de cet objet, et pas l'objet lui-même.

Par exemple :

```
>>> A, B = [1,2] # Cette syntaxe permet d'affecter le premier élément
de la liste à A et le second à B
>>> A
1
>>> B
2
>>> A = [1] # Ça ne marche plus quand la liste ne contient qu'un seul
élément : c'est la liste qui est affectée à A
>>> A
[1]
>>> A, = [1] # Avec cette syntaxe, c'est le contenu de la liste qui
est affecté à A.
>>> A
1
```

Les arguments de `FuncAnimation` sont : la figure à tracer, la fonction qui est appelée, le nombre d'images et l'intervalle de temps en ms entre deux images.

L'option `blit = True` (valeur par défaut : `False`) permet de n'actualiser que les parties qui ont changé.

L'option `repeat = True` (valeur par défaut : `True`) permet de recommencer en boucle l'animation, l'option `repeat_delay` donne alors le délai en ms entre la fin d'une animation et de début de la nouvelle.