

Propagation de paquets d'onde

I Rappels et contexte physique

De nombreux phénomènes ondulatoires présentent un comportement plus subtil que les vibrations d'une corde parfaite. Ils sont régis par des équations aux dérivées partielles linéaires qui admettent des solutions en OPPH

$$\underline{f}(x, t) = Ae^{i(kx - \omega t)}$$

à condition que k et ω soient liés par une relation de dispersion $\mathcal{R}(\omega, k) = 0$. Dans ce document, nous supposons que k est donné et que ω en dépend pour écrire la relation de dispersion sous la forme $\omega = \omega(k)$. Nous excluons tout phénomène d'absorption pour nous concentrer sur la dispersion et, par souci de simplicité, nous nous limitons à des OPPH se propageant vers les x croissants. Dans ces conditions, k et $\omega(k)$ sont des nombres réels de même signe.

La linéarité des équations d'onde concernées permet de considérer les OPPH comme des « briques » à partir desquelles on obtient, par superposition, des solutions plus générales :

$$\underline{f}(x, t) = \sum_n A_n e^{j[k_n x - \omega(k_n) t]} \quad (1)$$

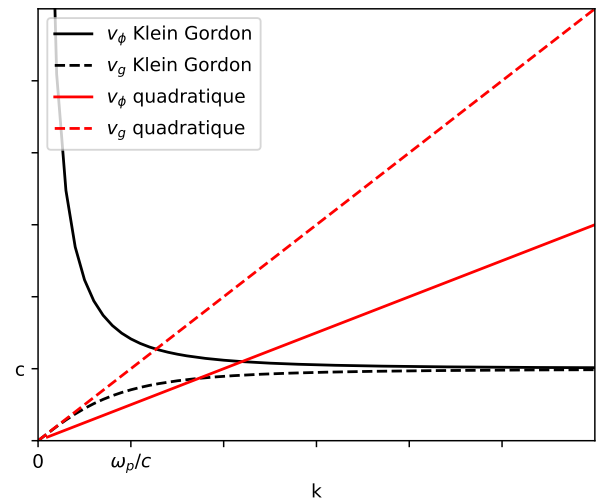
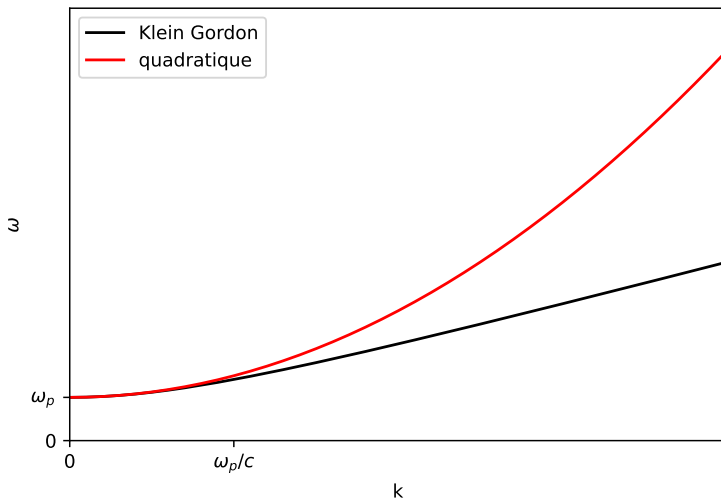
$$\text{ou} \quad \underline{f}(x, t) = \int_{-\infty}^{\infty} A(k) e^{j[kx - \omega(k) t]} dk \quad (2)$$

La somme discrète (1) peut comporter un nombre fini de termes, ou bien une infinité. Au travers des séries de Fourier, elle s'applique notamment à une onde périodique dans l'espace. La forme (2), plus générale et applicable à des fonctions non périodiques, fait intervenir toutes les pulsations spatiales $k \in \mathbb{R}$.

Ces notions s'appliquent à une multitude de situations mais dans la suite, nous nous limitons à deux relations de dispersion particulières :

- la relation de dispersion $\omega^2 = c^2 k^2 + \omega_p^2$ associée à l'équation de Klein-Gordon, et qu'on rencontre notamment dans l'étude de la propagation dans un plasma ;
- la relation de dispersion des ondes de de Broglie $\omega = \beta k^2$ avec $\beta = \hbar/(2m)$.

Pour ces deux relations de dispersion, les variations de ω , de la vitesse de phase v_ϕ et de la vitesse de groupe v_g en fonction de k sont représentées sur les figures ci-dessous.



II Paquets d'onde rudimentaires

Dans cette partie, nous construisons des paquets d'onde par application de la relation (1), en choisissant à notre guise les amplitudes A_n et les pulsations spatiales k_n . Les constantes physiques du problème (c , ω_p , β) sont traitées comme des variables globales.

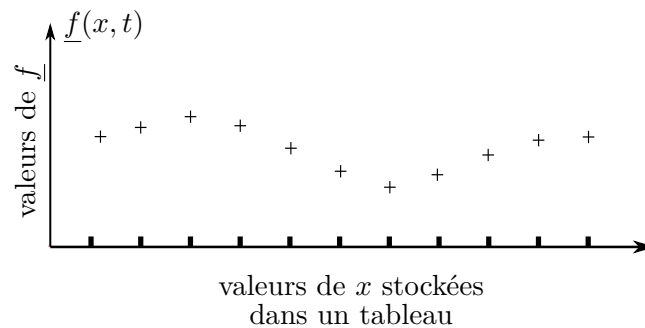
1. Accédez au fichier `paquet_ode_eleve.py` et faites-en une copie personnelle. Il contient des fonctions exprimant ω , v_ϕ et v_g en fonction de k pour les deux relations dispersion citées dans la partie précédente.

Dans la suite, pensez à utiliser les fonctions `dispersion_KG` et `dispersion_de_Broglie` !

2. La fonction $\underline{f}$ sera évaluée pour un nombre fini de valeurs équidistantes de x , stockées dans un tableau de numpy `x`. On obtient les valeurs de f par la somme (1), calculée au sein la fonction `somme_OPPH(A, k, x, t, fonction_omega)` prenant en paramètres :

- un tableau ou une liste `A` contenant les amplitudes A_n (éventuellement complexes) ;
- un tableau ou une liste `k` contenant les pulsations spatiales k_n ;
- un tableau `x` de flottants désignant un ensemble d'abscisses ;
- un flottant `t` désignant la date ;
- une fonction `fonction_omega` qui renvoie $\omega(k)$ (l'utilisateur pourra ainsi choisir à chaque appel une relation de dispersion particulière).

Cette fonction renvoie, sous la forme d'un tableau de numpy de nombres complexes, les valeurs de $\underline{f}(x, t)$, évaluée en chacune des abscisses du tableau `x`.



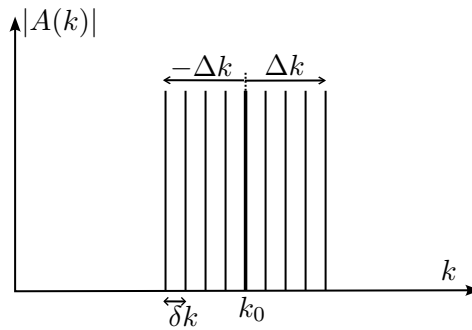
Pour la coder, on initialisera d'abord un tableau de numpy par `onde = np.zeros_like(x, dtype = complex)`, puis on calculera la somme (1) par une boucle. *Il est inutile de faire une autre boucle pour parcourir les différentes valeurs de x !* Ces abscisses sont représentées par un tableau de numpy, ce qui permet la vectorisation des opérations. Par exemple, `np.exp(x)` calcule l'exponentielle de toutes les valeurs contenues dans le tableau `x`, et les renvoie dans un nouveau tableau de numpy.

☞ Astuce : sous Python, le nombre complexe j s'écrit `1j` (sans rien entre 1 et J).

3. Utilisons pour l'instant la dispersion de Klein-Gordon. Comme première application, superposer deux OPPH de même amplitude et de longueurs d'onde $\lambda_1 = 5$ u et $\lambda_2 = 5,2$ u (u désigne une unité arbitraire).

- a. Pour 10000 valeurs de l'abscisse dans l'intervalle $[0, 300]$ u (`x = np.linspace(0, 300, 10000)`), calculer les valeurs de l'onde à $t = 0$, à $t = t_1 = 1$ u puis à $t = t_2 = 50$ u et stocker les résultats dans trois tableaux `onde_0`, `onde_t1`, `onde_t2`.
- b. Représenter ces « photographies » de l'onde en utilisant trois couleurs distinctes (`plt.plot(x, np.real(onde_0), color = 'ma_jolie_couleur')`).
- c. À partir du graphique, estimer la vitesse de phase et la vitesse de groupe. Pour la vitesse de phase, exploiter `onde_0` et `onde_t1` et utiliser un zoom. Pour la vitesse de groupe, exploiter `onde_0` et `onde_t2`.
- d. En utilisant les fonctions fournies, évaluer numériquement v_ϕ et v_g à la pulsation moyenne $k_0 = (k_1 + k_2)/2$.
- e. Vérifier que la valeur de $v_\phi(k_0)$ et de $v_g(k_0)$ sont en accord avec celles déduites du graphique.

4. On considère maintenant la superposition de $N = 2N_1 + 1$ ondes de pulsations $k_n = k_0 + i \delta k$ avec $i \in \{-N_1, -N_1 + 1, \dots, -1, 0, 1, \dots, N_1\}$. Le spectre forme un « peigne » à N dents ; il possède la pulsation centrale k_0 et la demi-largeur $\Delta k = N_1 \delta k$.



On utilise la relation de dispersion des ondes de de Broglie. On choisit $\lambda_0 = 5$ u, $k_0 = 2\pi/\lambda_0$, $\delta k = 0,02$ u, $A(k_n) = 1$ et $N = 11$ (c'est à dire $N_1 = 5$).

- Pour 1000 valeurs de x dans l'intervalle $[0, 200]$ u, calculer les valeurs de l'onde pour $t = 0$, $t = t_1 = 1$ u et $t = t_2 = 50$ u et stocker les résultats dans trois tableaux `onde_0`, `onde_t1`, `onde_t2`.
- Les représenter en utilisant trois couleurs distinctes.
- À $t = 0$, relever sur la courbe la largeur à mi-hauteur Δx de l'enveloppe et calculer le produit $\Delta x \times \Delta k$.
- En exploitant la courbe à t_2 , déterminer le déplacement d'enveloppe puis la comparer à la valeur attendue $v_g(k_0)$.
- En conservant la même valeur de δk , reprendre les calculs pour $N_1 = 10$ (c'est à dire $N = 21$, la largeur spectrale Δk augmente donc d'un facteur 2). Relever à nouveau Δx et calculer le produit $\Delta x \times \Delta k$. Commenter.

III Propagation depuis un état initial

Dans les exemples précédents, nous avons choisi de manière assez arbitraire les pulsations spatiales k_n et les amplitude A_n , ce qui donne à l'onde un aspect bien peu naturel. Il serait plus satisfaisant de considérer un train d'onde d'allure réaliste à $t = 0$ (figure 1) puis de déterminer comment il évolue au fil du temps. Comme l'onde n'est généralement pas périodique, nous utilisons la forme intégrale (2)

$$\underline{f}(x, t) = \int_{-\infty}^{\infty} A(k) e^{j[kx - \omega(k)t]} dk = \int_{-\infty}^{\infty} A(k) e^{j[k(x - v_\varphi(k))t]} dk \quad . \quad (3)$$

La condition initiale est définie par la fonction $\underline{f}_0(x) = \underline{f}(x, 0)$. En évaluant la relation ci-dessus à $t = 0$, on obtient

$$\underline{f}_0(x) = \int_{-\infty}^{\infty} A(k) e^{jkx} dk \quad . \quad (4)$$

Autrement dit, et bien que cela ne saute pas aux yeux, l'onde initiale est la somme d'une infinité de sinusoïdes. Les détails mathématiques qui sous-tendent ces idées ne figurent pas à votre programme, mais il est utile de connaître le vocabulaire suivant.

- La fonction $A : k \mapsto A(k)$ se somme *transformée de Fourier* de la fonction $\underline{f}_0 : x \mapsto \underline{f}_0(x)$. Quand on passe de $\underline{f}_0(x)$ à $A(k)$, on dit qu'on calcule une transformée de Fourier (TF en abrégé).
- Inversement, quand on passe de $A(k)$ à $\underline{f}_0(x)$ en appliquant la relation (4), on dit qu'on calcule une transformée de Fourier inverse.

$$\begin{array}{ccc} & \text{transformée} \\ & \text{de Fourier} \\ \underline{f}_0(x) & \xrightarrow{\quad} & A(k) \\ & \xleftarrow{\quad} & \\ & \text{transformée} \\ & \text{de Fourier inverse} \end{array}$$

Nous avons vu en cours que les largeurs utiles Δx et Δk des fonctions $\underline{f}_0(x)$ et $A(k)$ sont liées par

$$\Delta x \Delta k \geq \frac{1}{2} \quad .$$

Les relations (3) et (4) sont au cœur de notre raisonnement et montrent comment on peut, à partir d'une condition initiale $\underline{f}_0(x)$ donnée, obtenir l'onde pour $t > 0$. On procède pour cela en trois étapes.

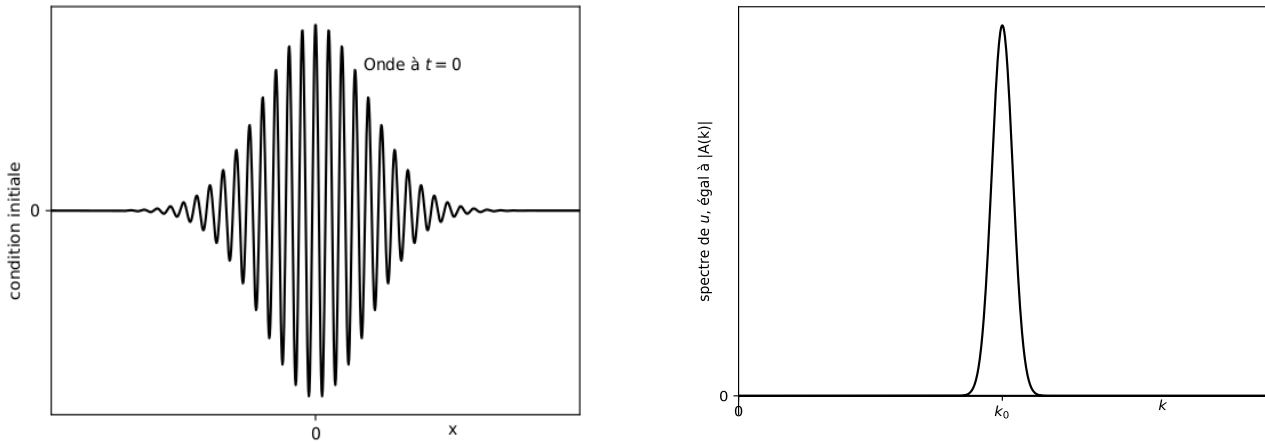


FIGURE 1 – Exemple de condition initiale et spectre associé

- (a) À partir de $\underline{f}_0(x)$, on calcule $A(k)$ par transformée de Fourier (vous n'êtes pas censés savoir le faire vous-même).
- (b) Chaque composante $A(k)e^{jkx}$ est l'évaluation d'une OPPH à $t = 0$. Elle se propage avec la vitesse $v_\varphi(k)$ et devient à $t > 0$:

$$A(k)e^{j[k(x-v_\varphi(k)t)]} = A(k)e^{jkx-j\omega(k)t} = A'(k)e^{jkx} \quad \text{avec} \quad A'(k) = A(k)e^{-j\omega(k)t}.$$

Le déplacement de cette composante entre $t = 0$ et $t > 0$ revient à lui faire subir en tout point un déphasage égal à $-\omega(k)t$.

- (c) À t , l'onde s'obtient en additionnant toutes ces OPPH selon la relation (3) qu'on réécrit

$$\underline{f}(x, t) = \int_{-\infty}^{\infty} A'(k)e^{jkx} dk \quad .$$

Autrement dit, $\underline{f}(x, t)$ est la transformée de Fourier inverse de $A'(k)$.

Ce schéma de pensée en trois étapes est représenté sur figure (2) qui exprime l'idée la plus importante de ce document ¹.

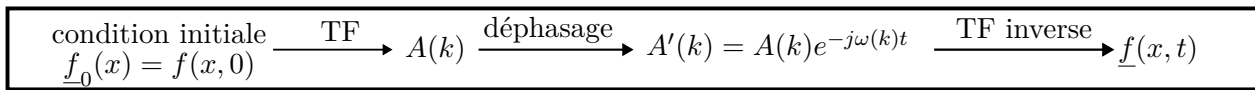


FIGURE 2 – Principe de propagation d'une onde par décomposition de Fourier

IV Passage à la transformée de Fourier discrète (TFD)

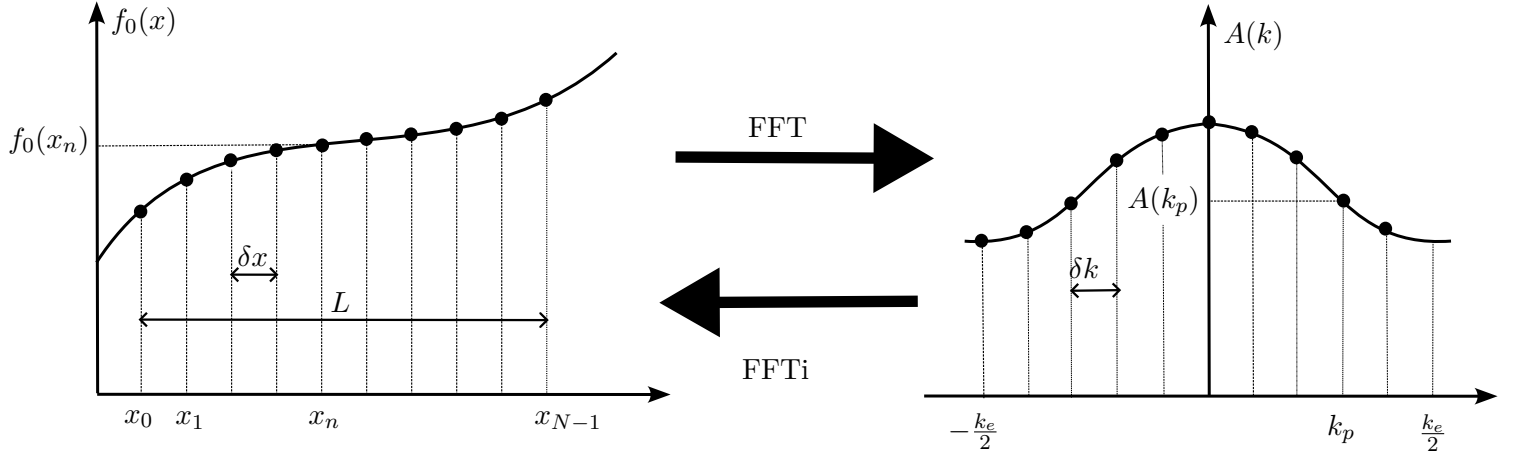
Numériquement, les fonctions $\underline{f}_0(x)$ et $\underline{f}(x, t)$ (pour t donné) ne sont pas calculées pour tout x , mais seulement pour N abscisses x_i équidistantes (N est supposé pair) couvrant un segment de largeur $L = N\delta x$. Ces abscisses sont

$$x_0 \quad , \quad x_1 = x_0 + \delta x \quad , \quad x_2 = x_0 + 2\delta x \quad , \quad \dots \quad , \quad x_{N-1} = x_0 + (N-1)\delta x \quad .$$

1. Signalons que l'expression (3), si elle convient à nos objectifs, n'est pas la plus générale. Pour k donné, la relation de dispersion admet en effet deux solutions ω , l'une positive et l'autre négative, et nous n'avons tenu compte que d'une seule. Nous nous limitons volontairement à des ondes se propageant vers la droite et par ailleurs, dans les exemples étudiés, $A(k)$ sera nulle pour $k < 0$. Dans la partie utile de l'intégrale (3), k est donc positif. Cela nous autorise à choisir systématiquement la solution $\omega > 0$ de la relation de dispersion.

De même, les pulsations k considérées sont au nombre de N . Elles sont séparées les unes des autres de $\delta k = \frac{2\pi}{L}$ et s'étendent de $-k_e/2$ (inclu) à $k_e/2$ (exclu), avec $k_e = N\delta k = 2\pi/(\delta x)$. Ces pulsations sont

$$-\frac{k_e}{2}, -\frac{k_e}{2} + \delta k, -\frac{k_e}{2} + 2\delta k, \dots, \frac{k_e}{2} - \delta k \quad .$$



Corrélativement à ce passage de variables continues à des variables discrètes, la transformée de Fourier est remplacée par une *transformée de Fourier discrète* (TFD). Dans les relations (3) et (4), on approxime les intégrales par des sommes selon la méthode des rectangles, pour écrire par exemple

$$\underline{f}_0(x_n) = \sum_{p=0}^{N-1} A(k_p) e^{ik_p x_n} \delta k \quad . \quad (5)$$

On voit que cette démarche se rapproche de celle adoptée dans la partie (II). Il y a cependant un énorme avantage à utiliser la TFD : elle peut être calculée très rapidement par transformée de Fourier rapide (FFT) au moyen du module `numpy.fft`. Nous utiliserons seulement trois fonctions de ce module.

- La fonction `np.fft.fft(f_0)` prend en paramètre le tableau contenant $\{f_0(x_n), n \in \{0, 1, \dots, N-1\}\}$ et renvoie celui contenant la TFD de f_0 , c'est à dire dans notre cas $\{A(k_p), p \in \{0, 1, \dots, N-1\}\}$. Cette fonction calcule le spectre échantillonné comme le font les oscilloscopes numériques².
- La fonction `np.fft.ifft(A)` prend en paramètre le tableau contenant des valeurs de $\{A(k_p), p \in \{0, 1, \dots, N-1\}\}$ et renvoie celui contenant $\{f_0(x_n), n \in \{0, 1, \dots, N-1\}\}$. Cela revient à synthétiser une fonction à partir de son spectre.

$$f_0(x_0), f_0(x_1), \dots, f_0(x_{N-1}) \xrightleftharpoons[\text{FFT inverse}]{\text{FFT}} A(k_0), A(k_1), \dots, A(k_{N-1})$$

- L'expression `2 * np.pi * np.fft.fftfreq(N, delta_x)` renvoie $\{k_p, p \in \{0, 1, \dots, N-1\}\}$, c'est à dire les pulsations spatiales pour lesquelles le spectre est évalué³. Il est important de comprendre que ces pulsations spatiales sont automatiquement fixées à partir des paramètres d'échantillonnage N de δx , avant même de considérer les valeurs de la fonction. On retrouve la même situation sur un oscilloscope numérique : il affiche le spectre sur tout un ensemble de fréquences respectant le critère de Shannon-Nyquist.

V Codage et observation de l'étalement

On considère les fonctions

$$f_{0\text{gauss}}(x) = A_0 e^{jk_0 x} e^{-\frac{x^2}{2\sigma_0^2}} \quad \text{et} \quad f_{0\text{tronq}}(x) = \begin{cases} A_0 e^{jk_0 x} & \text{si } |x| \leq \sigma_0 \\ 0 & \text{sinon.} \end{cases}$$

2. Pour être plus précis, la TFD calcule les $A(k_n)$ à un facteur près qui leur donne la dimension d'un coefficient de série de Fourier, mais ce facteur plus ou moins conventionnel ne joue aucun rôle dans la suite.

3. Pour de subtiles raisons, ces pulsations et les termes de la TFD sont rangés dans un ordre surprenant, les pulsations négatives apparaissant après les positives. On peut au besoin les réordonner avec `np.fft.fftshift`.

Elles représentent des paquets d'onde s'étendant sur une longueur de l'ordre de $2\sigma_0$ autour de l'origine, le premier avec une enveloppe gaussienne (comme sur la figure 1), le second en sinusoïde abruptement tronquée. Il est facile de montrer que la gaussienne présente la largeur à mi-hauteur $\ell_{1/2} = \sigma_0\sqrt{2\ln 2}$.

1. Coder les fonctions `onde_init_gaussienne(A0, x, sigma0, k0)` et `onde_init_tronquee(A0, x, sigma0, k0)` renvoyant dans un tableau numpy les valeurs de ces fonctions évaluées en tous les points du tableau `x`. Selon votre motivation, vous pouvez ne traiter qu'un seul de ces deux exemples.

2. On choisit $A_0 = 1$, $k_0 = 2\pi/5$ et $\sigma_0 = 20$. Sauf mention contraire, ces valeurs sont conservées dans la suite et traitées comme des variables globales. En prenant $N = 10000$, représenter ces fonctions pour bien visualiser ce que représente σ_0 .

3. En utilisant `np.fft.fftfreq` et `np.fft.fft`, calculer et représenter le spectre de ces ondes initiales. Le faire une première fois avec $\sigma_0 = 20$, puis une seconde fois avec $\sigma_0 = 10$. Comparer les deux situations et commenter.

Dans toute la suite, les pulsations spatiales échantillonnées, qui ont été calculées par `fftfreq` sont stockées dans un tableau `k` considéré comme une variable globale.

4. Coder la fonction `propage(onde_init, t, fonction_omega)` qui opère la propagation de l'onde en utilisant le principe de la figure (2). Elle prend en paramètres :

- un tableau `onde_init` contenant les N valeurs $f_0(x_n)$ de l'onde échantillonnée à $t = 0$;
- un flottant t représentant la date ;
- une fonction `fonction_omega` exprimant la relation de dispersion sous la forme $\omega(k)$.

Elle renvoie le tableau des N valeurs de $f(x_n, t)$.

Munis de ces outils, nous pouvons maintenant procéder à diverses applications et illustrer le comportement d'une onde dans un milieu dispersif. Dans les exemples qui suivent, je vous recommande de choisir `x = np.linspace(-100, 1000, N)` avec $N = 10000$.

5. On considère dans un premier temps un paquet d'onde gaussien et la relation de dispersion quadratique. Calculer et représenter sur une même figure l'onde initiale, puis à $t = 500u$. Calculer numériquement la vitesse de groupe. Relever sur le graphique le déplacement de l'enveloppe et le comparer à celui attendu.

6. Sur la figure précédente, observer *l'étalement du paquet d'onde* et répondre à la question : pourquoi le paquet d'onde s'étaile-t-il ? Un calcul exact montre que dans cet exemple particulier, la largeur à mi-hauteur de l'enveloppe évolue au fil du temps selon

$$\ell_{1/2}(t) = \ell_{1/2}(0) \sqrt{1 + \left(\frac{\beta v t}{\ell_{1/2}(0)^2} \right)^2} .$$

Mesurer la largeur à mi-hauteur à $t = 0$ puis à $t = 500u$, et confronter la valeur obtenue à cette prédiction.

7. Expliquer pourquoi on s'attend à ce que l'étalement soit d'autant plus rapide que σ_0 est plus petit.

8. Prendre une condition initiale en sinusoïde tronquée avec $\sigma_0 = 20$. Représenter l'onde à $t = 0$ et $t = 500u$. Analyser à nouveau le déplacement de l'enveloppe et observer l'étalement. Reprendre le calcul en réduisant σ_0 (par exemple $\sigma_0 = 5$) et commenter.

9. On souhaite maintenant observer l'influence de k_0 . Pour la dispersion gaussienne puis pour la dispersion quadratique, comparer l'étalement à $t = 500u$, selon que $k_0 = 2\pi/5$ ou $k_0 = 2\pi/10$ (cela fait donc 4 cas à étudier). Commenter les résultats en lien avec les courbes fournies à la fin de la partie (I).