1 Modélisation de signaux numériques

L'objectif de ce TP est de créer des signaux et d'observer leur éventuelle dispersion. En effet, lorsqu'une onde se propage sans atténuation, le spectre du signal porté par l'onde ne varie pas en amplitude, mais est déphasé d'une quantité qui dépend de la vitesse de l'onde et de la distance parcourue. Lorsque la vitesse d'une onde dépend de la fréquence, les composantes du spectre du signal n'arrivent pas toutes en même temps et le signal s'en retrouve modifié : c'est la dispersion.
Pour illustrer ce phénomène, nous allons programmé des signaux numériques en utilisant le langage de programmation *Python*. Pour chaque commande *cmd*, une aide peut être apportée en saisissant la commande *help(cmd)*. Dans un premier temps, il s'agira de définir des signaux ayant des spectres larges et dans un second temps, nous nous efforcerons à faire se propager les signaux sur différentes distances et différentes vitesses.

1.1 Définition des signaux

L'objectif de cette première partie est de définir et tracer des signaux numériques, ainsi que leur transformée de Fourier.

1. Importer les librairies *numpy* et *matplotlib.pyplot*.
2. Définir un vecteur temps t d'une durée $1000T$ avec $T = 0,01$ s à l'aide de la fonction *linspace* de la librairie *numpy*.
3. Créer un signal sinusoïdal $s(t) = \sin(2\pi f_0 t)$, de fréquence $f_0 = 100$ Hz.
4. Représenter le signal $s(t)$ à l'aide de la fonction *plot* de la librairie *matplotlib.pyplot*.
5. Calculer et représenter le module de la transformée de Fourier $S(\omega)$ du signal $s(t)$. Pour cela, il faudra utiliser les fonctions *fft* et *fftfreq* de la librairie *numpy.fft*
6. Calculer la transformée de Fourier inverse de $S(\omega)$ à l'aide de la fonction *ifft* de la librairie *numpy.fft* et vérifier que le résultat obtenu est égal à $s(t)$. Pour cela, il suffit de tracer les deux signaux à comparer sur une même figure.
7. Refaire les questions 4, 5 et 6 en considérant un signal défini par la relation :

$$s(t) = \sin(2\pi f_0 t) e^{-\frac{1}{2}[\sigma(t-t_0)]^2}. \tag{1}$$

   Prendre $\sigma = 300$ Hz et $t_0 = 20$ ms.

1.2 Propagation des signaux

L'objectif est maintenant d'observer la propagation du signal défini par la relation (1) avec différentes vitesses. Considérons un signal $s_1(t)$ de transformée de Fourier $S_1(\omega)$ en un point x_1 de l'espace. Si ce signal est porté par une onde se propageant à une vitesse de phase V, le signal au point x_2 est alors noté $s_2(t) = s_1(t - d/V)$, où $d = x_2 - x_1$ est la distance parcourue entre les deux points de mesure x_1 et x_2. Il vient alors que la transformée de Fourier $S_2(\omega)$ est donnée par l'expression

$$S_2(\omega) = S_1(\omega) e^{-ikd} \qquad \text{avec} \qquad k = \frac{\omega}{V}. \tag{2}$$

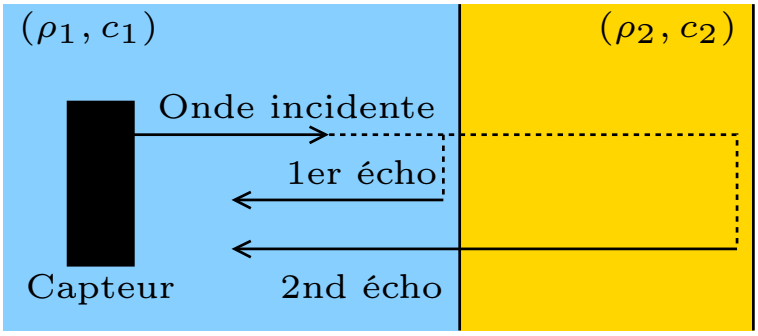
1. Calculer le signal $s_2(t)$ s'étant propagé sur une distance $d = 100$ mm et à une vitesse $V = 340$ m/s à l'aide de la relation (2).
2. Faire une boucle *for* afin de calculer le signal $s_2(t)$ sur quelques positions et représenter les signaux obtenus en fonction du temps.
3. Reprendre les deux premières questions en considérant une vitesse dépendant de la fréquence :

$$V = 3\sqrt{\omega} \tag{3}$$

4. Reprendre les questions précédentes en rajoutant du bruit sur le signal $s_1(t)$ d'amplitude de quelques % à l'aide de la fonction *rand* de la librairie *numpy.random*.

2 Traitement du premier écho : mesure de l'impédance

L'échographie ultrasonore est utilisée dans différents domaines, de l'imagerie médicale à l'acoustique sous-marine. Cette technique repose sur la génération d'ondes acoustiques dans le milieu à sonder et la détection des ondes réfléchies à chaque interface. L'objectif de ce TP est alors d'analyser des signaux en réflexion afin de déterminer la compressibilité et la masse volumique du matériau sondé. Un schéma de l'expérience est présenté sur la figure ci-contre. Le milieu 1 est de l'eau caractérisé par la masse volumique $\rho_1 = 1000$ kg/m^3 et la vitesse du son $c_1 = 1500$ m/s et le matériau est caractérisé par la masse volumique ρ_2 et la vitesse du son c_2, toutes les deux inconnues. Le capteur est déplacé selon une grille parallèle à l'échantillon inspecté et génère une onde de impulsion et détecte deux ondes : une onde réfléchie à la première interface entre l'eau et l'échantillon et une seconde qui correspond à la réflexion sur la face arrière de l'échantillon. La durée d'acquisition étant limitée, les échos suivant ne sont pas acquis.



2.1 Traitement du premier écho : mesure de l'impédance

1. Télécharger depuis moodle le fichier *Data_Signaux.h5* qui contient les signaux à traiter.
2. Télécharger depuis moodle le fichier *Echographie.py* qui permet de charger les signaux et les variables utiles au traitement.
3. Représenter le signal de référence *signal_ref* en fonction du temps t.

4. En utilisant la fonction *argmax* de la librairie *numpy*, déterminer le temps d'arrivée t_{ref} et l'amplitude A_{ref} du maximum du signal.

5. Utiliser un signal quelconque du tableau *Signal* et déterminer l'amplitude A_1 du maximum du premier écho. Pour rappel, cet écho arrive précisément à t_{ref}.

6. Calculer le coefficient de réflexion $r_1 = A_1/A_{ref}$.

7. Automatiser la procédure afin de visualiser la carte de l'impédance.

2.2 Traitement du second écho : mesure de la vitesse du son

Afin d'améliorer l'image de l'objet, le second écho est désormais traité.

1. Utiliser un signal quelconque du tableau *Signal* et le découper en deux signaux contenant chacun un des deux échos. Représenter les deux signaux obtenus.

2. En utilisant la fonction *argmax*, déterminer les temps d'arrivée t_1 et t_2 des deux premiers échos.

3. Sachant que l'épaisseur de l'échantillon est $h = 10$ mm, déterminer la vitesse du son c_2.

4. Automatiser la procédure afin de visualiser les cartographies de la densité ρ_2 et de la compressibilité $\chi_2 = 1/(\rho_2 c_2^2)$.

5. Trouver une échelle de couleur (colormap) adaptée à la compressibilité de l'objet sondé !

3 Propagation de la propagation acoustique dans un guide à section variable

L'objectif de ce TP est de modéliser la propagation d'ondes acoustiques dans un guide à section variable. Ce type de modélisation trouve son intérêt dans plusieurs domaines de l'acoustique : acoustique urbaine (propagation dans une rue), acoustique musicale (instruments à vents, exemples : trompette, flûte, ...), amplification du son (mégaphone, stéthoscope, ...).

3.1 Modélisation par la méthode de la matrice de transfert

L'objet de cette partie est de modéliser la propagation dans un guide à section variable composé de N couches de section S_n et d'épaisseur h_n. En amont ($x < 0$) et en aval ($x > x_N$) de ce milieu multicouche, les sections sont constantes et notées respectivement S_0 et S_{N+1} (figure 1).

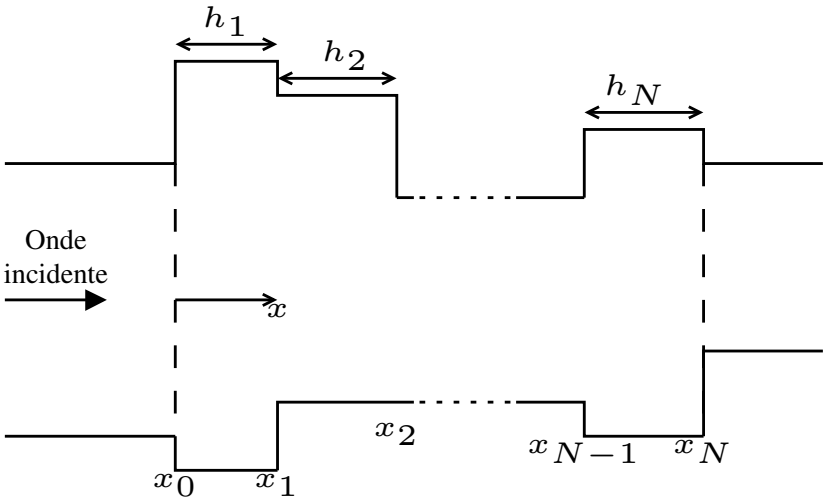


FIGURE 1 – Schéma du milieu multicouche.

La fréquence des ondes est supposée inférieure à la première fréquence de coupure f_1. Seul le mode plan peut donc être conservé dans la modélisation. Cela se traduit par une indépendance de la pression acoustique selon les coordonnées transverses du guide : la pression acoustique ne dépend alors que de la coordonnée axiale. Dans une couche n quelconque, la pression acoustique correspond à la superposition d'une onde progressive et d'une onde régressive :

$$p_n(x) = \left(A_n e^{ik_0 x} + B_n e^{-ik_0 x}\right) e^{-i\omega t}. \tag{4}$$

Compte tenu de l'équation de conservation de la quantité de mouvement, la vitesse normale a pour expression

$$v_n(x,t) = \frac{1}{Z_0}\left(A_n e^{ik_0 x} - B_n e^{-ik_0 x}\right) e^{-i\omega t}. \tag{5}$$

Le débit peut alors être mis sous la forme :

$$q_n(x,t) = \frac{S_n}{Z_0}\left(A_n e^{ik_0 x} - B_n e^{-ik_0 x}\right) e^{-i\omega t}. \tag{6}$$

Le vecteur d'état $\boldsymbol{\zeta}_n(z)$, défini à partir de la relation

$$\begin{pmatrix} p_n(x,t) \\ q_n(x,t) \end{pmatrix} = \boldsymbol{\zeta}_n(x) e^{-i\omega t}, \tag{7}$$

est donné par :

$$\boldsymbol{\zeta}_n(x) = \boldsymbol{D}_n \boldsymbol{E}_n \boldsymbol{X}_n \qquad \text{avec} \qquad \boldsymbol{D}_n = \begin{pmatrix} 1 & 1 \\ \dfrac{S_n}{Z_0} & -\dfrac{S_n}{Z_0} \end{pmatrix} \qquad \text{et} \qquad \boldsymbol{E}_n = \begin{pmatrix} e^{ik_0 x} & 0 \\ 0 & e^{-ik_0 x} \end{pmatrix} \tag{8}$$

et où $\boldsymbol{X}_n = (A_n, B_n)^T$. Cette dernière expression permet de relier le vecteur d'état au deux extrémités d'une couche :

$$\boldsymbol{\zeta}_n(x_{n-1}) = \boldsymbol{T}_n \boldsymbol{\zeta}_n(x_n) \qquad \text{avec} \qquad \boldsymbol{T}_n = \begin{pmatrix} \cos(k_0 h_n) & -i\dfrac{Z_0}{S_n}\sin(k_0 h_n) \\ -i\dfrac{S_n}{Z_0}\sin(k_0 h_n) & \cos(k_0 h_n) \end{pmatrix}. \tag{9}$$

Les pressions et vitesses normales étant continues à chaque interface, compte tenu des lois de Snell-Descartes, les vecteurs d'état sont également continus aux interfaces. Cela permet d'exprimer le vecteur d'état du milieu 0 en $x = 0$ en fonction de celui du milieu $N + 1$ à l'interface $x = x_N$:

$$\underline{\zeta}_0(x_0) = \prod_{n=1}^{N} \boldsymbol{T}_n \underline{\zeta}_{N+1}(x_N) \tag{10}$$

Compte tenu de l'expression (8), le vecteur d'état dans les deux demi-espaces sont définis par les expressions

$$\begin{cases} \boldsymbol{\zeta}_0(x=0) = \boldsymbol{D}_0 \boldsymbol{X}_0 & \text{avec} \quad \boldsymbol{X}_0 = (A, rA)^T, \\ \boldsymbol{\zeta}_{N+1}(x=x_N) = \boldsymbol{D}_{N+1} \boldsymbol{X}_{N+1} & \text{avec} \quad \boldsymbol{X}_{N+1} = (tA, 0)^T. \end{cases} \tag{11}$$

Dans ces deux expressions, A est l'amplitude arbitraire de l'onde incidente, r le coefficient de réflexion en amont du milieu multicouche et t le coefficient de transmission en aval. Compte tenu de ces expressions, l'équation matricielle (10) peut être mise sous la forme

$$\boldsymbol{X}_0 = \boldsymbol{M} \boldsymbol{X}_{N+1} \qquad \text{avec} \qquad \boldsymbol{M} = \boldsymbol{D}_0^{-1} \prod_{n=1}^{N} \boldsymbol{T}_n \boldsymbol{D}_{N+1}. \tag{12}$$

Les coefficients de transmission et de réflexion sont alors obtenus directement par les expressions

$$t = \frac{1}{\boldsymbol{M}_{11}} \qquad \text{et} \qquad r = \boldsymbol{M}_{21} t = \frac{\boldsymbol{M}_{21}}{\boldsymbol{M}_{11}}. \tag{13}$$

4 Modélisation numérique

L'objectif de ce TP est de développer un code numérique permettant le calcul des coefficients de réflexion r et de transmission t d'une structure en forme de guide à section variable. Le code doit pouvoir simuler la propagation dans n'importe quelle variation de section.

1. Définir deux vecteurs $\boldsymbol{S}$ et $\boldsymbol{h}$ à N éléments, où $\boldsymbol{S}$ contient les valeurs des sections et $\boldsymbol{h}$ leur épaisseur.
2. Définir une fonction calculant la matrice $\boldsymbol{T}_n$ en utilisant la *array* de la librairie *numpy*.
3. Créer une boucle pour pour calculer la matrice $\boldsymbol{M}$. Le produit de tableau se faire à l'aide de la fonction *dot* de la librairie *numpy*.
4. Calculer la coefficients r et t.
5. Généraliser les étapes précédentes en implémentant une boucle pour faire varier la fréquence.
6. Valider le code développé en comparant le coefficient de transmission obtenu avec celui d'une chambre à expansion (voir polycopié de cours).