Projet $n°1$ - 3 séances

Problèmes de restauration de signaux.

Dans ce problème nous voulons restaurer un son musical qui a été dégradé lors de son enregistrement.

Comme nous allons le voir la dégradation que nous considérons est supposée connue (par exemple par des expériences qui étudient les appareils concernés) et appartient à la famille des opérateurs dits de « convolution ». Ce même type de dégradation permet de modéliser par exemple les flous dus à un mouvement ou à un défaut de mise au point dans les photos.

Notre objectif est d'abord de comprendre le modèle mathématique pour décrire le signal dégradé et voir comment il est possible à partir de là de dérouler une stratégie pour restaurer au mieux le signal original.

1 Introduction

1.1 Son numérique

« Sous son aspect physique le son est un signal de pression liés aux éléments (groupes de molécules) du milieu où il se propage, ce milieu étant un gaz, un liquide ou un solide. [...] Quand le récepteur peut être l'oreille humaine, sensible aux vibrations acoustiques de fréquences comprises entre 20 Hz et 20 kHz, on parlera alors de vibrations sonores ou de sons. Le récepteur peut être également un capteur physique comme un microphone. »(citation d'un cours d'acoustique).

Mathématiquement on peut modéliser un son comme une fonction $F : t \mapsto F(t)$ à valeurs réelles, définie sur un intervalle $[0, T]$. Ce son peut être mesuré et enregistré. Pour être traité par un ordinateur il faut le numériser c'est à dire qu'à partir de F on construit un vecteur u dans $\mathbb{R}^N$ et on le fait classiquement de la manière suivante.

On choisit un intervalle de temps $t_s = T/N$ (N entier), qui indique à quelle cadence on va prendre les échantillons. On note $f_s = \frac{1}{t_s}$ qui est appelée **« cadence »** ou encore **« fréquence d'échantillonnage »**.

Le signal numérisé est alors obtenu en calculant pour $0 \leq n \leq N-1$ $u[n] = F(nt_s) = F\left(\frac{n}{f_s}\right) = F\left(\frac{nT}{N}\right)$. Le vecteur $u = (u[0], \ldots u[N-1]) = \left(F\left(\frac{0}{f_s}\right), \ldots, F\left(\frac{(N-1)}{f_s}\right)\right)$ correspond donc à la numérisation du signal F (voir figure 1).

On se rend compte sur cette figure 1 que si la cadence d'échantillonnage f_s est trop petite, le signal pourrait être dégradé au moment de la numérisation.

A partir du son numérisé nous pouvons travailler à l'aide d'algèbre linéaire et de mathématiques pour effectuer des transformations sur ce son, et dans les cas qui nous intéressent restaurer des sons qui ont été dégradés.

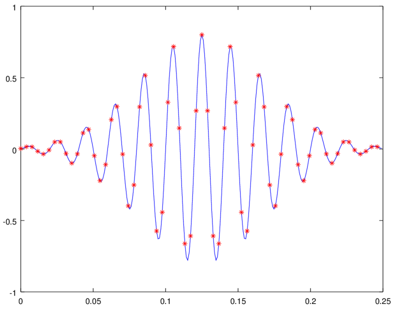
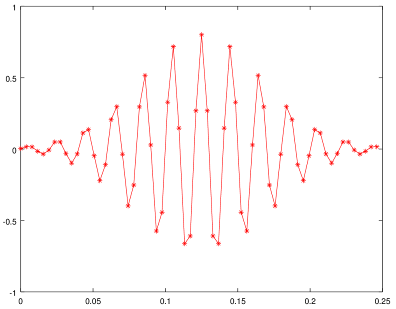

FIGURE 1 – A gauche : le signal original F représenté sur $[0, T]$ en bleu a été échantillonné et on a donc construit le vecteur $u = (u[0], \ldots u[N-1])$ dont les valeurs correspondent aux ordonnées des points rouges. A droite : on trace en les reliant par des segments de droites les points $(t_n, u[n])$ avec $t_n = \frac{n}{f_s}$.

Question 1: Avec Python

Soit $F : t \mapsto \cos(2\pi\omega_0 t)e^{-\frac{(t-0.5)^2}{2\sigma^2}}$ où ω_0 est un entier et t varie dans $[0, 1]$, et $\sigma \in \mathbb{R}$.
On prend dans ce qui suit $\omega_0 = 20$ et $\sigma = 0.1$.

1. Tracer une représentation graphique de la fonction F.
2. Superposer sur la première figure le tracé des points $(t_n, u[n])$ en les reliant par des segments de droite en choisissant une première valeur de f_s.
3. Tracer le même type de figure pour une deuxième valeur de f_s, puis une troisième. Que constatez vous si $f_s << 2\omega_0$?

1.2 Dégradations

On vous a transmis un extrait sonore correspondant à un enregistrement qui s'est mal passé (micros défectueux).

Vous écouterez pendant la séance le son original, et sa version dégradé.

On note ainsi

1. u le vecteur de $\mathbb{R}^N$ qui est la version numérisée du son original. Il correspond au son numérique non dégradé auquel nous n'avons pas accès (sauf pour l'écouter en cours !) et que nous voulons restaurer.
2. y le vecteur de $\mathbb{R}^N$, le son dégradé. On modélise y comme la transformation de u par un opérateur supposé connu (à l'aide de mesures) $\mathcal{H} : \mathbb{R}^N \to \mathbb{R}^N$ de dégradation, c'est à dire

$$y = \mathcal{H}(u) \tag{1}$$

Nous avons besoin de la définition suivante

Définition 1 *Un opérateur linéaire $\mathcal{H} : \mathbb{R}^N \mapsto \mathbb{R}^N$ est un opérateur de convolution si il existe $h \in \mathbb{R}^N$ noté $h = (h_0, h_1, \ldots, h_{N-1})$ tel que la matrice de $\mathcal{H}$ dans la base canonique s'écrive*

$$H = \begin{pmatrix} h_0 & h_{N-1} & h_{N-2} & h_{N-3} & \ldots & h_2 & h_1 \\ h_1 & h_0 & h_{N-1} & h_{N-2} & \ldots & h_3 & h_2 \\ h_2 & h_1 & h_0 & h_{N-1} & & h_4 & h_3 \\ \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ h_{N-2} & h_{N-3} & \ldots & \ldots & \ldots & h_0 & h_{N-1} \\ h_{N-1} & h_{N-2} & \ldots & \ldots & \ldots & h_1 & h_0 \end{pmatrix}$$

Un opérateur de convolution est aussi appelé « opérateur de filtrage »en traitement mathématique du signal.
Le vecteur h est alors appelé réponse impulsionnelle de l'opérateur de filtrage $\mathcal{H}$.

On peut considérer pour notre problème que $\mathcal{H}$ est un opérateur linéaire de $\mathbb{R}^N$ dans $\mathbb{R}^N$ et qu'il fait partie de la classe des opérateurs de convolution sur $\mathbb{R}^N$ suivant la définition 1. Sa réponse impulsionnelle est le vecteur h de $\mathbb{R}^N$ qui est connu grâce à des mesures.

Vous avez ainsi à votre disposition

1. y sous forme de fichier `.wav`
2. h sous forme de fichier texte

Le but est de restaurer u.

Question 2: sur Ametice

1. Télécharger les données sur le site Ametice du cours ou le cas échéant sur la page web https://www.i2m.univ-amu.fr/perso/clothilde.melot/modelisation.
2. Transformer les données en vecteurs de type `Numpy`.

1.3 Problème à résoudre

Le vecteur y est donné dans la base canonique de $\mathbb{R}^N$.
La restauration de u revient donc à la résolution du système linéaire écrit dans la base canonique et d'inconnue x

$$y = Hx \tag{2}$$

On prendra alors $u = x$ comme solution de notre problème.

Le vecteur y étant de taille $\sim 10^6$ pour 23 secondes de son, la matrice H contient quant à elle 10^{12} coefficients.
Python ne peut pas stocker une telle matrice ! *Faites-en vous même l'expérience.*

1.4 Diagonalisation de l'opérateur $\mathcal{H}$

Nous allons cependant pouvoir montrer que l'opérateur $\mathcal{H}$ a pour matrice une matrice diagonale dans une autre base et qu'il est donc possible de travailler dans cette autre base pour résoudre le système (2).

2 Restauration du son

Nous avons besoin des notations suivantes. Soit $N \in \mathbb{N}$.

- On note ω le nombre $\omega = e^{\frac{2i\pi}{N}}$.
- On considère la famille de vecteurs $\mathcal{E} = \{\varepsilon^\ell \in \mathbb{C}^N, \ell = 0, \dots, N-1\}$ tels que pour $\ell = 0, \dots, N-1$ et pour $n \in \{0, \dots N-1\}$ la n-ième coordonnée du vecteur ε^ℓ s'écrive
$$\varepsilon_n^\ell = \omega^{ln} = e^{\frac{2i\pi\ell n}{N}}$$
- On indice les lignes et les colonnes des matrices sur $\mathbb{C}^N$ de 0 à $N-1$.
- On note également $P = (p_{i,j})_{i,j=0..N-1}$ la matrice dont les vecteurs colonnes sont les vecteurs de $\mathcal{E}$ et P^* la matrice de coefficients $\tilde{p}_{i,j} = \overline{p_{j,i}}$.
- Enfin pour $a \in \mathbb{C}^N$ on écrit $\hat{a} = P^* a$. Pour $\ell = 0, \dots, N-1$, $\hat{a}_\ell$ est la ℓ-ième coordonnée de $\hat{a}$ dans la base canonique de $\mathbb{C}^N$.

2.1 Étude du cas $N = 3$

Nous prenons dans toute cette partie $N = 3$.

Nous allons commencer par étudier la matrice H en dimension 3 et montrer qu'elle est diagonalisable.
Nous considérons donc $\mathcal{H}$ un opérateur de convolution de réponse impulsionnelle $h \in \mathbb{R}^3$.

Avec les notations précisées au début de cette section on a $\hat{h} = P^*h$ et $\hat{h}_0, \hat{h}_1, \hat{h}_2$ sont les coordonnées de $\hat{h}$ dans la base canonique de $\mathbb{C}^3$.

La matrice H représentative de l'opérateur $\mathcal{H}$ dans la base canonique s'écrit $H = \begin{pmatrix} h_0 & h_2 & h_1 \\ h_1 & h_0 & h_2 \\ h_2 & h_1 & h_0 \end{pmatrix}$ et on considère la matrice $J = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$.

On a donc avec $N = 3$: $\omega = e^{\frac{2i\pi}{3}}$.

Question 3: en Python ou avec Latex

1. Calculer $1 + \omega + \omega^2$ et ω^3.
2. Calculer J^3.
3. En déduire les valeurs propres complexes de J.
4. Montrer qu'une base de vecteurs propres de J est constituée de la base $\mathcal{E}$.
5. Montrer que H peut s'écrire comme une combinaison linéaire des matrices Id, J et J^2 et en déduire les valeurs propres notées $\{\hat{h}_0, \hat{h}_1, \hat{h}_2\}$ de H. Donner une base de vecteurs propres associée.

   On note D la matrice diagonale $D = \begin{pmatrix} \hat{h}_0 & 0 & 0 \\ 0 & \hat{h}_1 & 0 \\ 0 & 0 & \hat{h}_2 \end{pmatrix}$ et on suppose dans toute la suite que D est inversible.
6. Calculer PP^* et en déduire P^{-1} en fonction de P^*.
7. Soient x et y tels que (2) soit vérifiée.
   Calculer $\hat{y} = P^*y$ en fonction de $\hat{x} = P^*x$, et de D.
8. Calculer la solution x du système (2) en fonction de $\hat{y}$, P, N et D^{-1}.

2.2 Étude du cas général

Ici on prend N entier quelconque supérieur ou égal à deux.

Nous commençons par montrer que $\mathcal{E}$ est bien une base de $\mathbb{C}^N$ qui est orthogonale pour le produit scalaire $\langle ., . \rangle$ canonique défini pour tout $x = (x_i)_{i=0,\ldots,N-1} \in \mathbb{C}^N$ et $y = (y_i)_{i=0,\ldots,N-1} \in \mathbb{C}^N$:

$$\langle x, y \rangle = \sum_{n=0}^{N-1} x_n \overline{y_n}$$

Proposition 1 *La famille $\mathcal{E}$ est une famille finie à N éléments qui forme une base orthogonale de $\mathbb{C}^N$ pour le produit scalaire $\langle ., . \rangle$.*

Éléments de démonstration de la proposition 1 :
On calcule les produits scalaires $\langle \varepsilon^\ell, \varepsilon^k \rangle$ pour $\ell \neq k$ après avoir démontré le lemme suivant

Lemme 1 *Soit $k_0 \in \mathbb{Z}$ et $N \in \mathbb{N}^\star$. Alors*

- *si* $k_0 = 0(modN)$ *on a* $\sum_{n=0}^{N-1} \omega^{k_0 n} = N$,
- *sinon* $\sum_{n=0}^{N-1} \omega^{k_0 n} = 0$.

On démontre enfin le théorème suivant.

Théorème 1 *Soit $\mathcal{H}$ un opérateur de convolution de réponse impulsionnelle $h \in \mathbb{R}^N$.*

Alors

1. *La base $\mathcal{E}$ est une base de diagonalisation de $\mathcal{H}$, c'est à dire que la matrice représentative D de $\mathcal{H}$ dans la base $\mathcal{E}$ est diagonale.*
2. *D a pour coefficients diagonaux les éléments $\hat{h}_\ell$ pour $\ell = 0, \dots, N-1$.*
3. *Soit $x \in \mathbb{R}^N$ un vecteur colonne et $y = Hx$. Alors*
   (a) *$\hat{y} = D\hat{x}$*
   (b) *on a aussi $y = Hx = \frac{1}{N} PD\hat{x}$*

Éléments de démonstration du théorème 1 :
Soit $\mathcal{H} : \mathbb{C}^N \mapsto \mathbb{C}^N$ un opérateur de convolution (ou de filtrage). On note h la réponse impulsionnelle de $\mathcal{H}$ et $\hat{h} = P^* h$. Plusieurs voies sont possibles. On peut par exemple procéder de la manière suivante.

1. On généralise la technique utilisée dans le cas $N = 3$ ou on écrit ε^ℓ comme un vecteur colonne et on calcule $H\varepsilon^\ell$ en utilisant le fait que $\omega^{ln} = \omega^{ln-N}$.
2. Le calcul précédent donne le résultat.
3. (a) On remarque que $P^{-1} = \frac{1}{N} P^*$. On a donc que $\hat{y} = D\hat{x}$.
   (b) Le résultat suit grâce à celui de la question précédente.

Remarque : Les calculs de $\hat{y}$ et $\hat{h}$ ne sont pas effectués tels quels par un ordinateur.
En effet un calcul direct donnerait en réalité pour chaque coordonnée de $\hat{y}$ N multiplications suivies de $N-1$ additions, soit un coût d'environ $2N^2$ opérations pour le calcul de $\hat{y}$ (*Pourquoi ?*). Dès que N commence à être grand (par exemple ici $N = 10^6$, ce qui est une taille très courante en image ou même en son), le calcul direct devient très coûteux, voire impossible à mener.

Heureusement dans le cas où on a $N = 2^{n_0}$ avec $n_0 \in \mathbb{N}^*$, nous avons la possibilité d'utiliser un algorithme basé sur des factorisations matricielles astucieuses pour effectuer le calcul de $\hat{y}$ à partir de y. Cet algorithme est appelé « Transformée de Fourier rapide »ou encore FFT en anglais (Fast Fourier Transform) et c'est un des algorithmes les plus efficaces du monde. Il permet de calculer $\hat{y}$ en $\mathcal{O}(N \ln(N))$ opérations, c'est à dire que le nombre d'opérations pour effectuer le calcul est proportionnel à $N \ln(N)$..

Le même type d'algorithme nous permettra de calculer la formule de reconstruction de (3b).

Question 4: Avec Python : La restauration du son !

En utilisant les résultats du théorème 1 restaurer avec Python le son $y = Hx$ pour obtenir $u = x$.

3 Routines Python pour résoudre ces problèmes

- importation d'un fichier `musique.wav` en un vecteur `y` (en conservant l'information sur la fréquence d'échantillonnage `fs`)

```
import scipy.io.wavfile as wavfile
fs, y = wavfile.read('musique.wav')
```

- importation d'un fichier texte `donnee.txt` en un vecteur `d`

```
import numpy as np

d = np.loadtxt('donnee.txt')
```

- importation d'un vecteur `y` en un fichier `musique.wav` (à l'aide de l'information sur la fréquence d'échantillonnage `fs`)

```
import numpy as np
import scipy.io.wavfile as wavfile

m=np.max(np.abs(y))
y=y/(1.1*m) # pour garantir que les coordonnees du vecteur sont entre -1 et 1.

wavfile.write('musique.wav',fs,y)
```

On désigne par `ychap` le vecteur $\hat{y}$.

- calcul de `ychap` à partir du vecteur `y`

```
import numpy.fft as npft

ychap=npft.fft(y) # les valeurs de ychap sont complexes.
```

- calcul de `y` à partir du vecteur `ychap`

```
import numpy.fft as npft

y=npft.ifft(ychap)

# les valeurs de y sont complexes en general apres cette operation dans tous
    les cas, a cause par exemple des erreurs d'arrondi.

# pour forcer le cas echeant y a etre reel (pour reconstruire un son numerique)
    on ajoute
y=y.real
```

4 Finalisation du projet

Le projet fera l'objet d'une soutenance orale (non notée) et son contenu mathématique et informatique seront au programme de l'examen terminal.

Il vous est conseillé de rédiger proprement les élements suivants (par exemple à l'aide de Latex ou de notebook) et de demander son avis sur ce document à l'enseignant qui s'occupe de votre groupe :

- Une description motivée des applications proposées. (Vous pourrez chercher dans la littérature ou sur internet des explications plus détaillées sur la numérisation du son, les problèmes de déconvolution...) ;
- Une représentation graphique des données proposées ;
- Une présentation des techniques mathématiques et informatiques utilisées. Vous répondrez aux questions posées en détaillant les raisonnements mathématiques le cas échéant ;
- Une illustration numérique de ces techniques, par exemple :
  - Illustrer numériquement le fait que la matrice H est inversible, par exemple en visualisant le module des coordonnées du vecteur $\hat{h}$. *Pourquoi cela permet de vérifier que la matrice H est inversible ?*
  - **Pour aller plus loin :**
    - Compléter les éléments de démonstration des résultats de la section 2.2.
    - On peut comprendre en détail l'algorithme de Transformée de Fourier rapide (FFT).

Références

[1] M. Bergounioux *Mathématiques pour le traitement du signal*, Dunod.

[2] B. Vallette Polycopié *Algèbre linéaire pour tous* https://www.math.univ-paris13.fr/~vallette/download/Alg%C3%A8bre%20lin%C3%A9aire%20pour%20tous.pdf