Représentations parcimonieuses des signaux et images

Parcimonie des opérateurs, application à la déconvolution

Dans ce problème nous voulons restaurer des sons musicaux qui ont été dégradés lors de leur enregistrement.

Comme nous allons le voir les dégradations que nous considérons sont supposées connues (par des expériences qui étudient les appareils concernés) et appartiennent 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 les signaux dégradés et voir comment il est possible à partir de là de dérouler une stratégie pour restaurer au mieux les signaux originaux.

Cette stratégie sera basée sur le fait que les opérateurs de dégradation ont une matrice parcimonieuse dans une base particulière.

Ils seront donc faciles à manipuler à la fois théoriquement et numériquement.

Les démonstrations des résultats dont l'énoncé commencent par (*) doivent être connues et feront l'objet d'évaluations.

Les données sont à télécharger lors de la première séance sur la page web http://www.i2m.univ-amu.fr/perso/clothilde.melot/parcimonie_m2ds ou sur le site Ametice du cours.

1 Sons dégradés

On vous a transmis deux extraits sonores correspondant à des enregistrements qui se sont mal passés (micros défectueux).

Vous écouterez pendant la première séance les sons originaux, et leurs versions dégradés.

On note ainsi

1. u le vecteur de $\mathbb{R}^N$ qui est la version numérisée du premier 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 $\mathbb{R}^N$, correspondant au premier 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

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

3. v le vecteur de $\mathbb{R}^N$ qui est la version numérisée du deuxième son original. De même que u il correspond au son numérique non dégradé auquel nous n'avons pas accès (sauf pour l'écouter en cours !) et que nous voulons restaurer.

4. z le vecteur de $\mathbb{R}^N$, correspondant au deuxième son dégradé. On modélise z comme la transformation de v par un opérateur supposé connu $\mathcal{G} : \mathbb{R}^N \to \mathbb{R}^N$ de dégradation, c'est à dire

$$z = \mathcal{G}(v) \tag{2}$$

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 que

1. $\mathcal{H}$ et $\mathcal{G}$ sont des opérateurs linéaires de $\mathbb{R}^N$ dans $\mathbb{R}^N$.
2. $\mathcal{H}$ et $\mathcal{G}$ font partie de la classe des opérateurs de convolution sur $\mathbb{R}^N$ suivant la définition 1. Leurs réponses impulsionnelles sont respectivement les vecteurs h et g de $\mathbb{R}^N$ qui sont connus grâce à des mesures.

Vous avez ainsi à votre disposition

1. y et h
2. z et g

Le but est de restaurer u et v.

2 Restauration du premier son

2.1 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

$$y = Hx \tag{3}$$

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.*

2.2 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 (3).

Nous allons donc montrer les trois résultats suivants en précisant d'abord quelques notations.

On note

- $\langle .,. \rangle$ le produit scalaire canonique sur $\mathbb{C}^N$ tel que $\langle x, y \rangle = \sum_{n=0}^{N-1} x_n \overline{y_n}$ pour $x = (x_i)_{i=0,\dots,N-1} \in \mathbb{C}^N$ et $y = (y_i)_{i=0,\dots,N-1} \in \mathbb{C}^N$.
- $\|x\|^2 = \sum_{n=0}^{N-1} |x_n|^2$ pour $x \in \mathbb{C}^N$
- Soit $A = (a_{i,j})_{i,j=1,\dots,N}$ une matrice de taille $N \times N$ à coefficients dans $\mathbb{C}$. On appelle adjoint de A la matrice A^* telle que
  $$\text{pour tout } a \text{ et tout } b \text{ dans } \mathbb{C}^N \ \langle Aa, b \rangle = \langle a, A^* b \rangle.$$
  La matrice A^* est donc la matrice de coefficient sur la ligne i et la colonne j $c_{i,j} = \overline{a_{j,i}}$.
- la famille de vecteurs $\mathcal{E} = \{e^\ell \in \mathbb{C}^N, \ell \in \mathbb{Z}\}$ tels que pour $\ell \in \mathbb{Z}$ et pour $n \in \{0, \dots N-1\}$ la n-ième coordonnée du vecteur e^ℓ s'écrive
  $$e^\ell_n = e^{\frac{2i\pi\ell n}{N}}$$
  Pour $x \in \mathbb{C}^N$ et $\ell \in \mathbb{Z}$ on note $\hat{x}_\ell = \langle x, e^\ell \rangle$.
  Enfin pour $x \in \mathbb{C}^N$ on note $\hat{x}$ le vecteur dont les coordonnées sont $\hat{x}_\ell$ pour $\ell \in \{0, \dots, N-1\}$.

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 $\langle .,. \rangle$. Elle est appelée base de Fourier finie.

Éléments de démonstration : • On remarque d'abord que $e^\ell = e^{N-\ell}$ et on a donc seulement N éléments distincts dans l'ensemble $\{e^\ell, \ell \in \mathbb{Z}\}$.

- On calcule les produits scalaires $\langle e^\ell, e^k \rangle$ 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 $k_0 = 0 (mod N)$ on a* $\sum_{n=0}^{N-1} e^{2i\pi \frac{nk_0}{N}} = N$,

— *sinon* $\sum_{n=0}^{N-1} e^{2i\pi \frac{nk_0}{N}} = 0$.

d'où la démonstration de la proposition 1. ■

La proposition permet de démontrer le corollaire suivant.

Corollaire 1

(*) Soit $x \in \mathbb{R}^N$ et $\hat{x}$ le vecteur dont les coordonnées sont $\hat{x}_\ell = \langle x, e^\ell \rangle$. Alors

1. *pour tout $\ell \in \{0, \dots, N-1\}$* $\hat{x}_\ell = \sum_{k=0}^{N-1} x_k e^{-\frac{2i\pi k\ell}{N}}$
2. $x = \frac{1}{N} \sum_{\ell=0}^{N-1} \hat{x}_\ell e^\ell$.
3. $\|x\|^2 = \frac{1}{N} \sum_{\ell=0}^{N-1} |\langle x, e^\ell \rangle|^2 = \frac{1}{N} \|\hat{x}\|^2$

Éléments de démonstration : 1. On développe l'expression de $\hat{x}_\ell$ à l'aide de e^ℓ_k.

2. On sait que $\{e^\ell, \ell = 0, \dots, N-1\}$ est une base de $\mathbb{C}^N$. Donc pour tout $x \in \mathbb{C}^N$ il existe $a_0, \dots, a_{N-1}$ tels que

$$x = \sum_{k=0}^{N-1} a_k e^k \tag{4}$$

Nous voulons montrer que pour tout $\ell = 0, \dots, N-1$ $a_\ell = \frac{\langle x, e^\ell \rangle}{N}$ et pour cela nous calculons $\langle x, e^\ell \rangle$ grâce à (4).

3. On développe $\|x\|^2 = \langle x, x\rangle$ à l'aide de la formule du point 2. ■

Remarque : La transformée $x \mapsto \hat{x}$ est appelée « Transformée de Fourier discrète »(ou DFT en anglais : Discrete Fourier Transform).

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) *pour tout $\ell \in \{0, \dots, N-1\}$ on a $\hat{y}_\ell = \hat{h}_\ell \hat{x}_\ell$.*
   (b) *on a aussi $y = Hx = \frac{1}{N} \sum_{\ell=0}^{N-1} \hat{h}_\ell \hat{x}_\ell e^\ell$*

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

1. On écrit e^ℓ comme un vecteur colonne et on calcule He^ℓ.
2. Le calcul précédent donne le résultat.
3. (a) Plusieurs méthodes sont là encore possibles. On peut par exemple exprimer les coordonnées de x et y dans la base $\mathcal{E}$ à l'aide par exemple du corollaire 1 ou bien si on préfère utiliser la matrice F de la base $\mathcal{E}$ qui a pour vecteurs colonnes les vecteurs e^ℓ.

   Comme D est la matrice représentative de $\mathcal{H}$ dans $\mathcal{E}$ le résultat suit.

   (b) On applique par exemple le corollaire 1. ■

Remarque : Les calculs de $\hat{x}$ et $\hat{h}$ ne sont pas effectués tels quels par un ordinateur.
En effet un calcul direct donnerait en réalité pour chaque coordonnée de $\hat{x}$ N multiplications suivies de $N-1$ additions, soit un coût d'environ $2N^2$ opérations pour le calcul de $\hat{x}$ (*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 $\hat{x}$ à partir de x. 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 $\hat{x}$ 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).

3 Restauration du deuxième son

Appliquons la stratégie précédente au cas du deuxième son. Il semble que cela ne marche pas. *Pourquoi ?*

Nous ne sommes plus en mesure de garantir que nous allons pouvoir reconstruire le son original ! C'est ce qui arrive aussi dans les problèmes réels ! !
Ce que nous pouvons néanmoins faire c'est développer une stratégie pour calculer une estimation la meilleure possible de ce son original.

En particulier remarquons que résoudre le système d'inconnue x

$$z = Gx \tag{5}$$

revient à calculer x tel que

$$\|Gx - z\|^2 = \min_{a \in \mathbb{R}^N} \|Ga - z\|^2 \tag{6}$$

L'avantage d'écrire la formulation (6) c'est qu'on peut toujours calculer x même si (5) n'a pas de solution.

En effet dans notre cas nous avons la proposition suivante (qui peut se généraliser avec un peu de travail à toute matrice non nulle).

Proposition 2

Soit $\mathcal{G}$ un opérateur de convolution de réponse impulsionnelle g non nulle, et G sa matrice dans la base canonique. Soit $z \in \mathbb{R}^N$.

Soit F la fonction de $\mathbb{R}^N$ dans $\mathbb{R}$ telle que

$$F(a) = \|Ga - z\|^2$$

Alors le problème

$(\mathcal{P})$ *Trouver $x \in \mathbb{R}^N$ tel que*

$$F(x) = \min\{F(a) : a \in \mathbb{R}^N\}$$

a au moins une solution dans $\mathbb{R}^N$.

Éléments de démonstration : Le problème ($\mathcal{P}$) est un problème dit de « moindres carrés ». Il y a plusieurs manières de le résoudre.

N'hésitez pas à utiliser ici vos connaissances !

Dans notre cas une stratégie possible est de le résoudre de façon explicite de la manière suivante.

Soit $\mathcal{G}$ un opérateur de convolution de réponse impulsionnelle g non nulle. On note G sa matrice dans la base canonique. Soit $a \in \mathbb{R}^N$ et $z \in \mathbb{R}^N$.

On a d'après ce qu'on a fait précédemment

$$F(a) = \frac{1}{N}\|\widehat{Ga} - \hat{z}\|^2$$

On peut alors calculer l'ensemble des solutions du problème ($\mathcal{P}$).

Dans le cas où $\hat{g}_\ell \neq 0$ pour tout ℓ et donc G inversible on retrouve la solution unique du système $Gx = z$. ■

Lorsque G n'est pas inversible le problème ($\mathcal{P}$) a une infinité de solutions et il est difficile de savoir laquelle serait la plus pertinente pour notre problème ! De plus il est aussi difficile de savoir laquelle est la plus proche de la solution recherchée v.

Une stratégie est alors de chercher à résoudre un problème proche de ($\mathcal{P}$), qui ne permettra pas de retrouver exactement v, mais qui a une solution unique. C'est le cas du problème suivant.

On fixe $\lambda > 0$ et on s'intéresse à l'application $F_\lambda : \mathbb{R}^N \to \mathbb{R}$ telle que $F_\lambda(a) = F(a) + \lambda\|a\|^2$. On examine alors le problème suivant

$(\mathcal{P}_\lambda)$ Trouver $x_\lambda \in \mathbb{R}^N$ tel que

$$F_\lambda(x_\lambda) = \min\{F_\lambda(a) : a \in \mathbb{R}^N\}$$

Proposition 3

Soit $\mathcal{G}$ un opérateur de convolution de réponse impulsionnelle g non nulle, et G sa matrice dans la base canonique. Soit $z \in \mathbb{R}^N$.

Alors le problème $(\mathcal{P}_\lambda)$ a une unique solution x dans $\mathbb{R}^N$ telle que pour tout $\ell \in \{0, \ldots, N-1\}$

$$\hat{x}_\ell = \frac{\overline{\hat{g}_\ell}\hat{z}_\ell}{|\hat{g}_\ell|^2 + \lambda}$$

Éléments de démonstration : La fonction F_λ est strictement convexe et différentiable sur $\mathbb{R}^N$, elle est de plus coercive. Donc elle a un unique minimum global en x tel que le gradient de F_λ s'annule en x.

On obtient donc que x le minimum vérifie

$$G^\star G x + \lambda x = G^\star z$$

Or remarquons que G^* a pour valeurs propres $\overline{\hat{g}_\ell}$ associées aux vecteurs propres e^ℓ pour $\ell = 0, \ldots, N-1$.

Donc $G^*G + \lambda I$ où I est la matrice identité a pour valeurs propres $|\hat{g}_\ell|^2 + \lambda$.

D'où le résultat. ■

4 Routines pour résoudre ces problèmes

Votre groupe doit choisir le langage avec lequel il veut travailler : soit avec R, soit avec Python.

4.1 Routines R

- importation d'un fichier `musique.wav` en une classe du type Wave `musique` à l'aide de par exemple la bibliothèque `seewave`, puis extraction du vecteur `y` qui correspond au canal de gauche (cas d'un son mono et non stereo)

```
library(seewave)

musique <- readWave("musique.wav")

# pour recuperer le canal de gauche, le seul pertinent si le son n'est pas un
    son stereo (ce qui est notre cas)
y <- musique@left
```

- importation d'un vecteur `y` en un fichier mono (non stereo) `musiquemod.wav`

```
musiquemod <- musique # on s'appuie sur le fichier de depart pour recuperer les
     donnees qui permettent de coder le son (frequence d'echantillonnage, etc
    ...)

musiquemod@left<- y
```

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

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

```
ychap <- fft(y) # les valeurs de ychap sont complexes.
```

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

```
y<- 1/N*Re(fft(ychap, inverse=TRUE))

# 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 <- Re(y)
```

4.2 Routines Python

- 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 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
```

5 Rendu du projet

Le projet consiste en

- Soit un fichier `notebook` codé en Python ou en R
- Soit un compte-rendu tapé en Latex dans un fichier .pdf et les codes associés. Dans ce cas il est indispensable que parmi les fichiers informatiques on trouve pour chaque partie pratique un fichier dont le titre permette de comprendre de quoi il s'agit et qu'il suffit d'executer pour illustrer informatiquement ce qui est indiqué dans le compte-rendu.

L'objectif du projet est la résolution détaillée des deux problèmes de restauration.

En particulier on n'oubliera pas de détailler

- la présentation des techniques mathématiques utilisées (Vous compléterez les éléments de preuves fournis dans l'énoncé et répondre aux questions *Pourquoi ?*) ;
- les illustrations numériques de ces techniques, par exemple :
  - l'illustration numérique du corollaire 1. En particulier on pourra chercher une interprétation physique de l'indice ℓ.
  - l'illustration numérique du fait que la matrice H est inversible et que la première méthode ne fonctionne pas pour la restauration du deuxième son

- **Pour aller plus loin :**
  - On peut considérer le cas où $\mathcal{H}$ et $\mathcal{G}$ sont des endomorphismes quelconques sur $\mathbb{R}^N$. Les résultats démontrés dans le sujet sont-ils encore valides ? Vous pouvez alors démontrer que c'est bien le cas sous certaines hypothèses. Cela permet d'envisager de résoudre d'autres problèmes que des problèmes de déconvolution. On peut ainsi s'intéresser au cas où $\mathcal{H}$ est un opérateur de masquage, c'est à dire que sa matrice dans la base canonique est une matrice diagonale qui n'a que des 0 ou des 1 sur la diagonale.
  - On peut comprendre en détail l'algorithme de Transformée de Fourier rapide (FFT).

Annexe

Compléments sur les sons numériques

« Sous son aspect physique le son est un ébranlement élastique des 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 $u = (u[0], \dots u[N-1]) = \left(f\left(\frac{0}{f_s}\right), \dots, f\left(\frac{(N-1)}{f_s}\right)\right)$ 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.

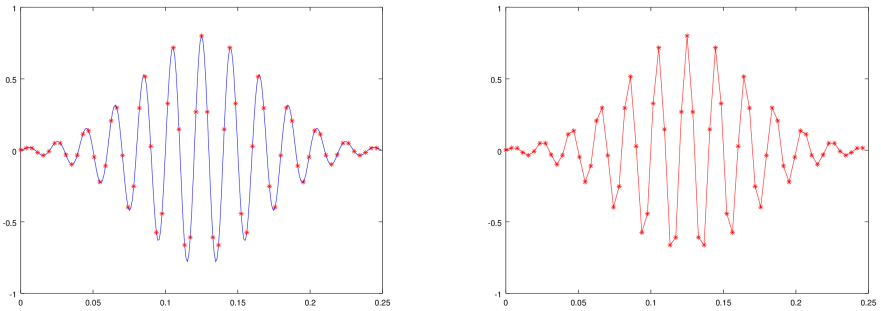

FIGURE 1 – A gauche : le signal original f représenté sur $[0, T]$ en bleu a été échantillonné et on a donc construit le vecteur $u = (u[0], \dots u[N-1])$ dont les valeurs correspondent aux ordonnées des points rouges. A droite : on trace en les reliant par des segments de droites les points $(t_n,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

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.

Références

[1] Philippe G. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Dunod.

[2] Bruno Galerne *Optimisation*, Polycopié de cours disponible à l'adresse http://w3.mi.parisdescartes.fr/~bgalerne/m1_optimisation/poly_optimisationv1.pdf

[3] Martin Vetterli, Jelena Kovacevic, Vivek K. Goyal. *Foundations of Signal Processing.* Academic Press,