



CY TECH

RAPPORT

---

# Projet Compressed Sensing

---

*Élèves :*

Ayoub EL HOUDRI  
Mlamali SAIDSALIMO

*Enseignant :*

M. Abderrahim  
BOURHATTAS

# Table des matières

<b>1</b>	<b>Représentation parcimonieuse dans une base orthonormée</b>	<b>2</b>
1.1	La base de Fourier discrète $F$	2
1.2	Représentation parcimonieuse d'un signal $N=500$	3
1.2.1	Construction d'un signal	3
1.2.2	Représentation parcimonieuse directe dans la base $C$	3
1.2.3	Représentation parcimonieuse directe dans la base $F$	5
1.2.4	Représentation parcimonieuse par OMP dans la base $C$	6
1.2.5	Comparaison des résultats	6
1.3	Représentation parcimonieuse d'un signal $N=100$	8
1.3.1	Construction d'un signal	8
1.3.2	Représentation parcimonieuse directe et par OMP dans la base $C$ et comparaison des résultats	9
1.3.3	Représentation parcimonieuse directe et par OMP dans la base $F$ et comparaison des résultats	11
1.4	Algorithmes de type OMP et dictionnaires formant une base	13
<b>2</b>	<b>Algorithme IRLS</b>	<b>13</b>
2.1	Equivalence de deux problèmes de minimisation	13
2.2	Solution d'un problème des moindres carrés	13
2.3	Représentation parcimonieuse d'un signal par IRLS	14
<b>3</b>	<b>Apprentissage d'un dictionnaire (k-SVD)</b>	<b>15</b>
3.1	Description des données ("DonneesProjet.xlsx")	15
3.2	Application de la k-SVD aux données	15
3.3	Comparaison des 3 Algorithmes MP, OMP et IRLS	15
<b>4</b>	<b>Matrices de mesure</b>	<b>20</b>
<b>5</b>	<b>Conclusion</b>	<b>22</b>

# 1 Représentation parcimonieuse dans une base orthonormée

## 1.1 La base de Fourier discrète F

On a :  $F_{p,q} = \frac{1}{\sqrt{N}} e^{-\frac{2pq}{N}i\pi}$  ( $0 \leq p, q \leq N - 1$ )

Montrons que la base formée par les  $F_{\bullet,q}$  ( $0 \leq p, q \leq N - 1$ ) est orthonormée.

$$\begin{aligned} C_{p,q} &= \langle F, F \rangle = F^T F \\ &= \sum_{n=0}^{N-1} \frac{1}{\sqrt{N}} e^{-\frac{2pn}{N}i\pi} \frac{1}{\sqrt{N}} e^{-\frac{2qn}{N}i\pi} \\ &= \sum_{n=0}^{N-1} \frac{1}{\sqrt{N}} e^{\frac{2pn}{N}i\pi} \frac{1}{\sqrt{N}} e^{-\frac{2qn}{N}i\pi} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} e^{\frac{2n(p-q)}{N}i\pi} \end{aligned}$$

Si  $p=q$  :

$$\begin{aligned} C_{p,q} &= \frac{1}{N} \sum_{n=0}^{N-1} 1 \\ &= \frac{N}{N} = 1 \end{aligned} \tag{1}$$

Si  $p \neq q$  :

$$\begin{aligned} C_{p,q} &= \frac{1}{N} \frac{1 - e^{\frac{2(p-q)N}{N}i\pi}}{1 - e^{\frac{2(p-q)}{N}i\pi}} \\ &= \frac{1}{N} \frac{1 - e^{2(p-q)i\pi}}{1 - e^{\frac{2(p-q)}{N}i\pi}} \\ &= \frac{1}{N} \frac{1 - 1}{1 - e^{\frac{2(p-q)}{N}i\pi}} = 0 \end{aligned} \tag{2}$$

Donc de (1) et (2) on conclut que  $C_{p,q} = \delta_{p,q}$

CCL :

Donc la base formée par les  $(F_{\bullet,q})_{0 \leq q \leq N-1}$  est orthonormée.

## 1.2 Représentation parcimonieuse d'un signal $N=500$

### 1.2.1 Construction d'un signal

On génère respectivement la porteuse et le signal sinus pour  $N = 500$  :

$$\begin{aligned} p(t) &= \cos(2\pi f_p t) & (f_p = 100) \\ s(t) &= \sin(2\pi f_0 t) & (f_0 = 50) \end{aligned}$$

Ainsi on construit le signal  $x$  tel que :

$$x(t) = s(t)p(t) = \sin(2\pi f_0 t) \cos(2\pi f_p t)$$

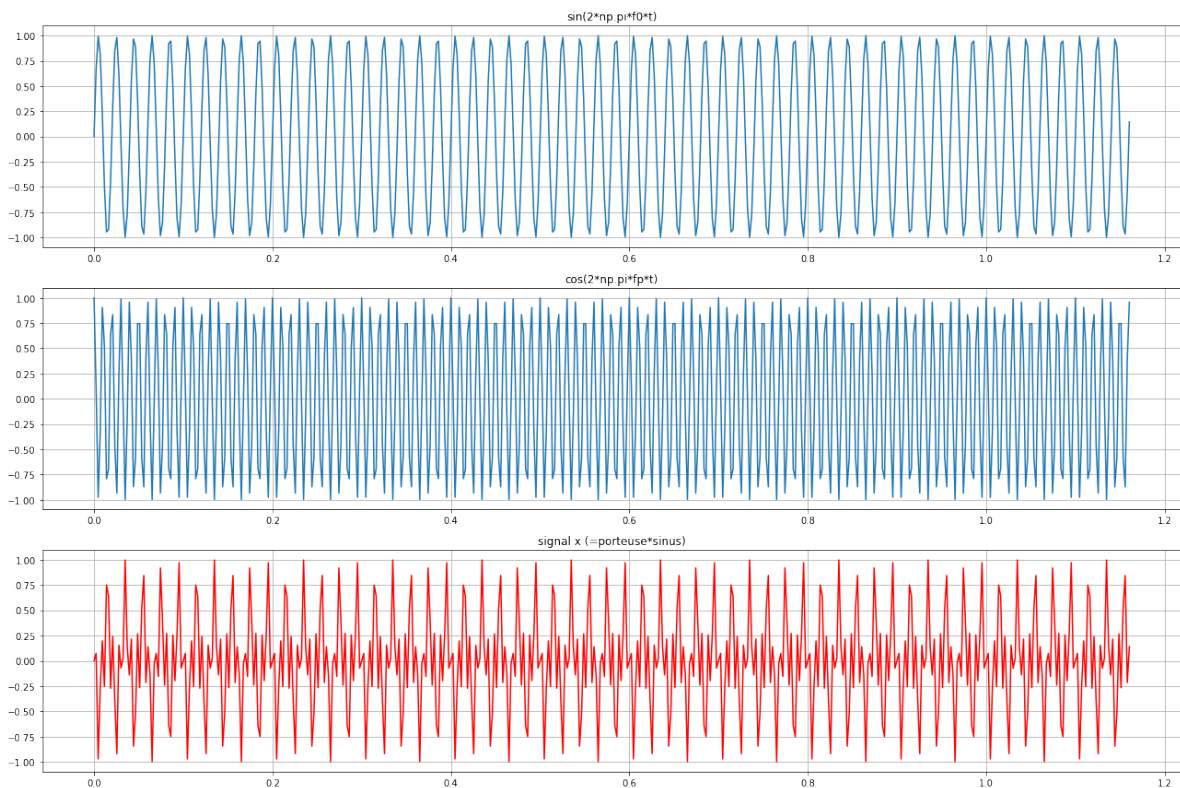


Figure 1 : Signal  $s$ , porteuse  $p$  et signal construit  $x$

### 1.2.2 Représentation parcimonieuse directe dans la base $C$

La représentation parcimonieuse directe du signal  $x$  dans la base  $C$ , la base de transformation en cosinus discrète (DCT) est donnée par la formule suivante :

$$\alpha_C = C^T x$$

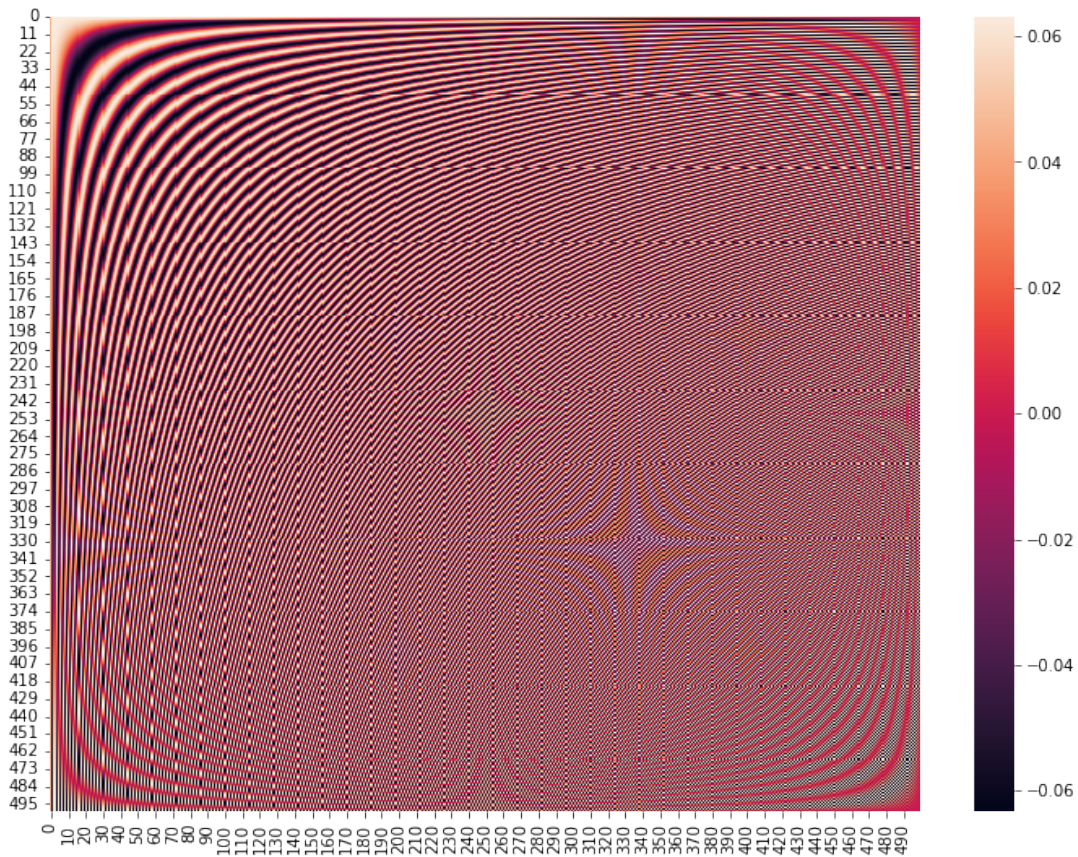


Figure 2 : Affichage de la matrice  $C$

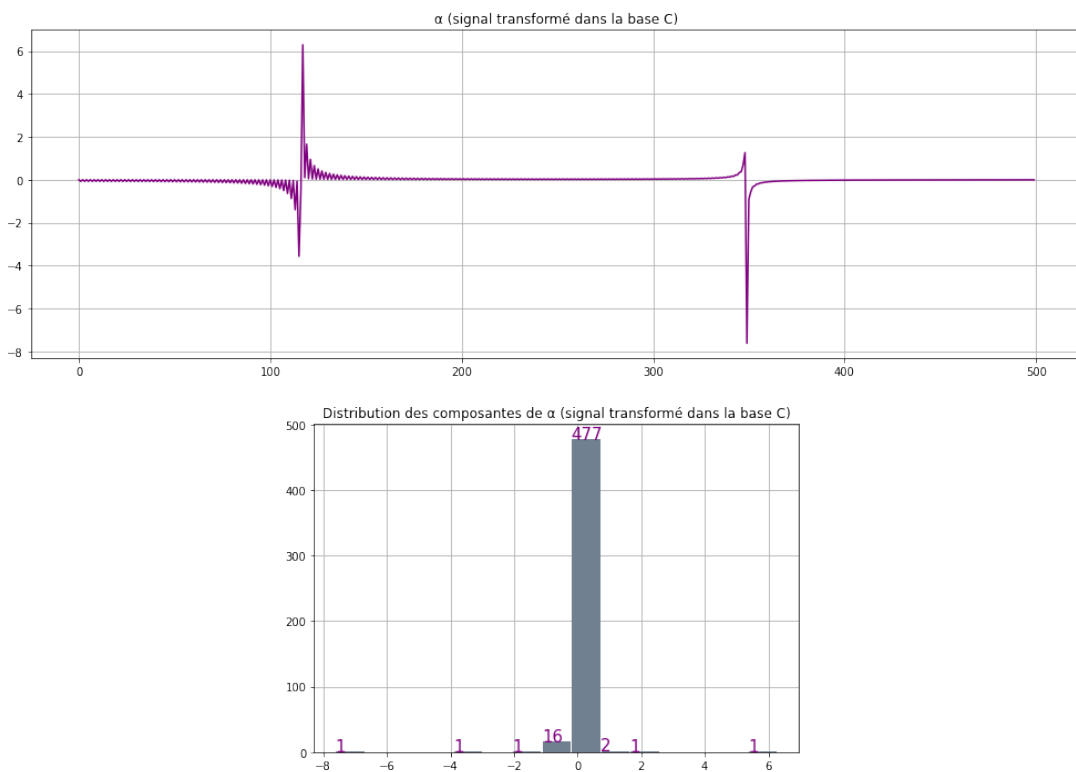


Figure 3 : Représentation parcimonieuse  $\alpha_C$  et histogramme des coefficients de  $\alpha_C$

### 1.2.3 Représentation parcimonieuse directe dans la base $F$

De même la représentation parcimonieuse directe du signal  $x$  dans la base  $F$ , la base de Fourier discrète (DFT) est donnée par la formule suivante :

$$\alpha_F = F^T x$$

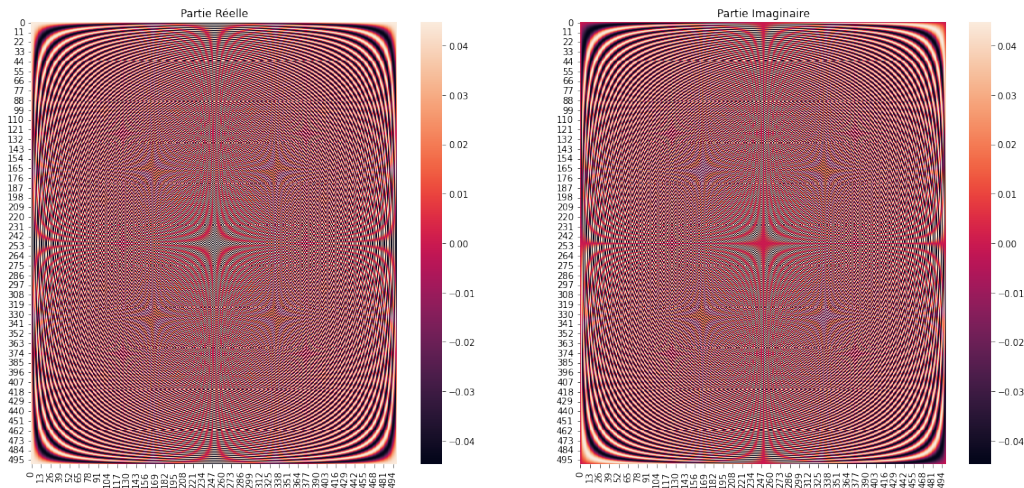


Figure 4 : Affichage de la partie réelle et imaginaire de la matrice  $F$

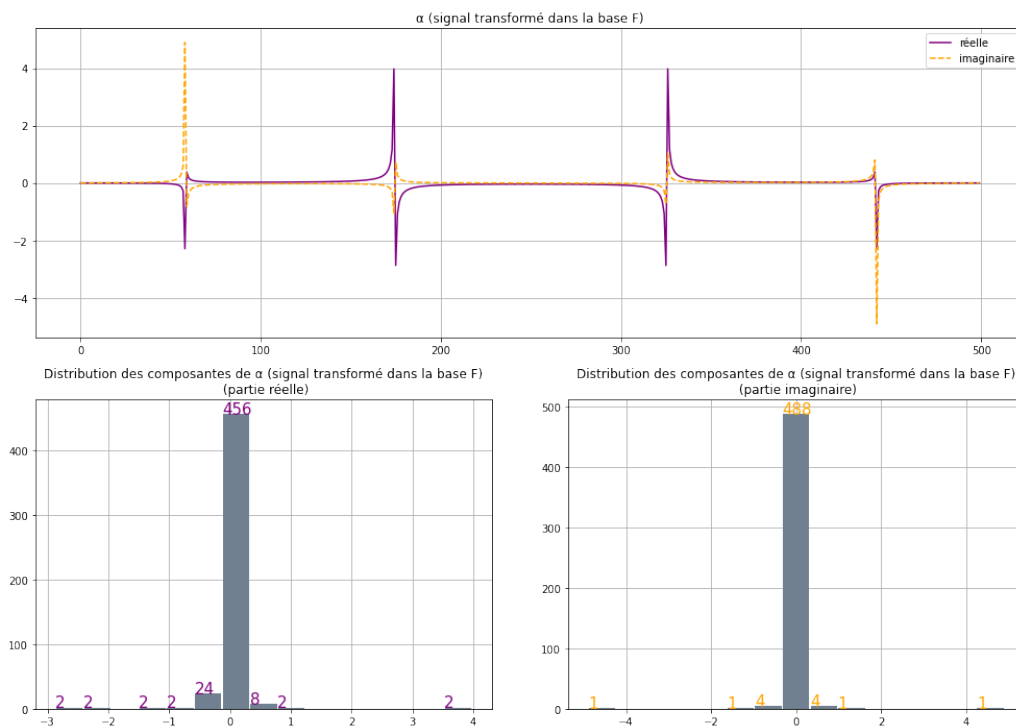


Figure 5 : Représentation parcimonieuse  $\alpha_F$  et histogramme des coefficients de  $\alpha_F$

### 1.2.4 Représentation parcimonieuse par OMP dans la base C

En appliquant l'algorithme OMP (Open Matching Pursuit) au signal  $x$  avec les paramètres suivants :

$$\begin{aligned} \epsilon &= 10^{-4} \\ IterMax &= 1000 \\ x &= \sin(2\pi f_0 t) \cos(2\pi f_p t) \\ D &= C \end{aligned}$$

On obtient le résultat suivant :

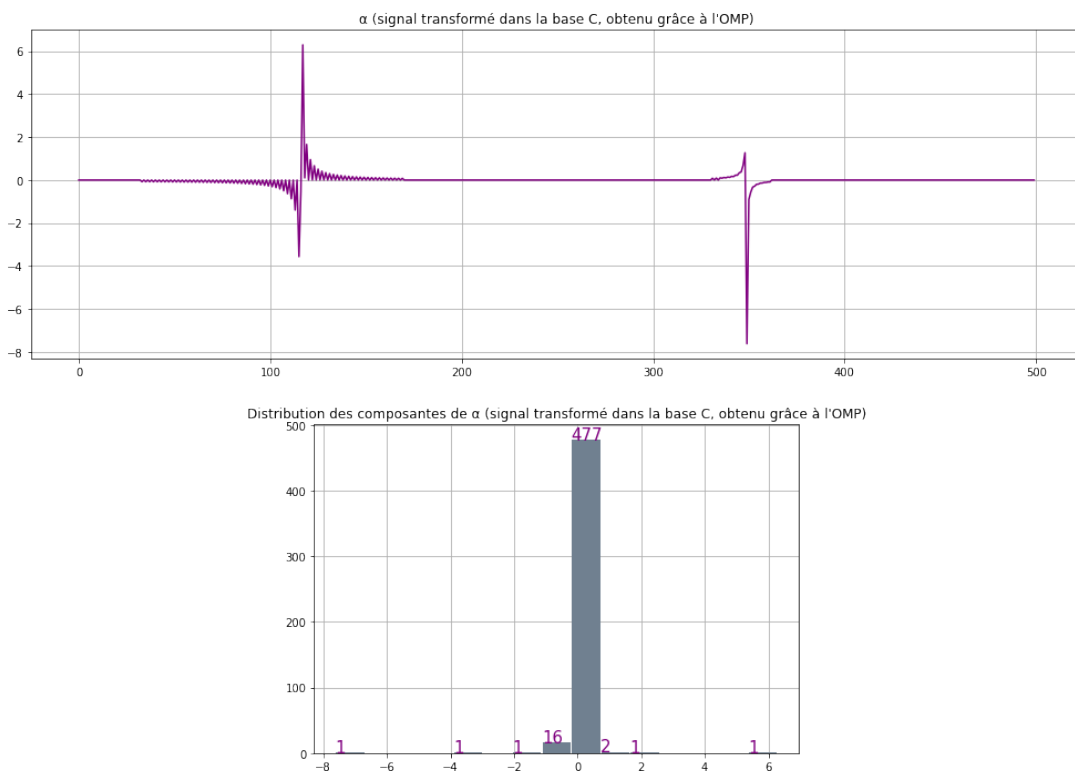


Figure 6 : Signal  $x$  et sa représentation parcimonieuse  $\alpha_{OMP_C}$  par OMP dans la base  $C$

### 1.2.5 Comparaison des résultats

Les visualisations ci-dessous montrent que les deux représentations parcimonieuses  $\alpha_C$  et  $\alpha_{OMP_C}$  du signal  $x$  sont presque les mêmes.

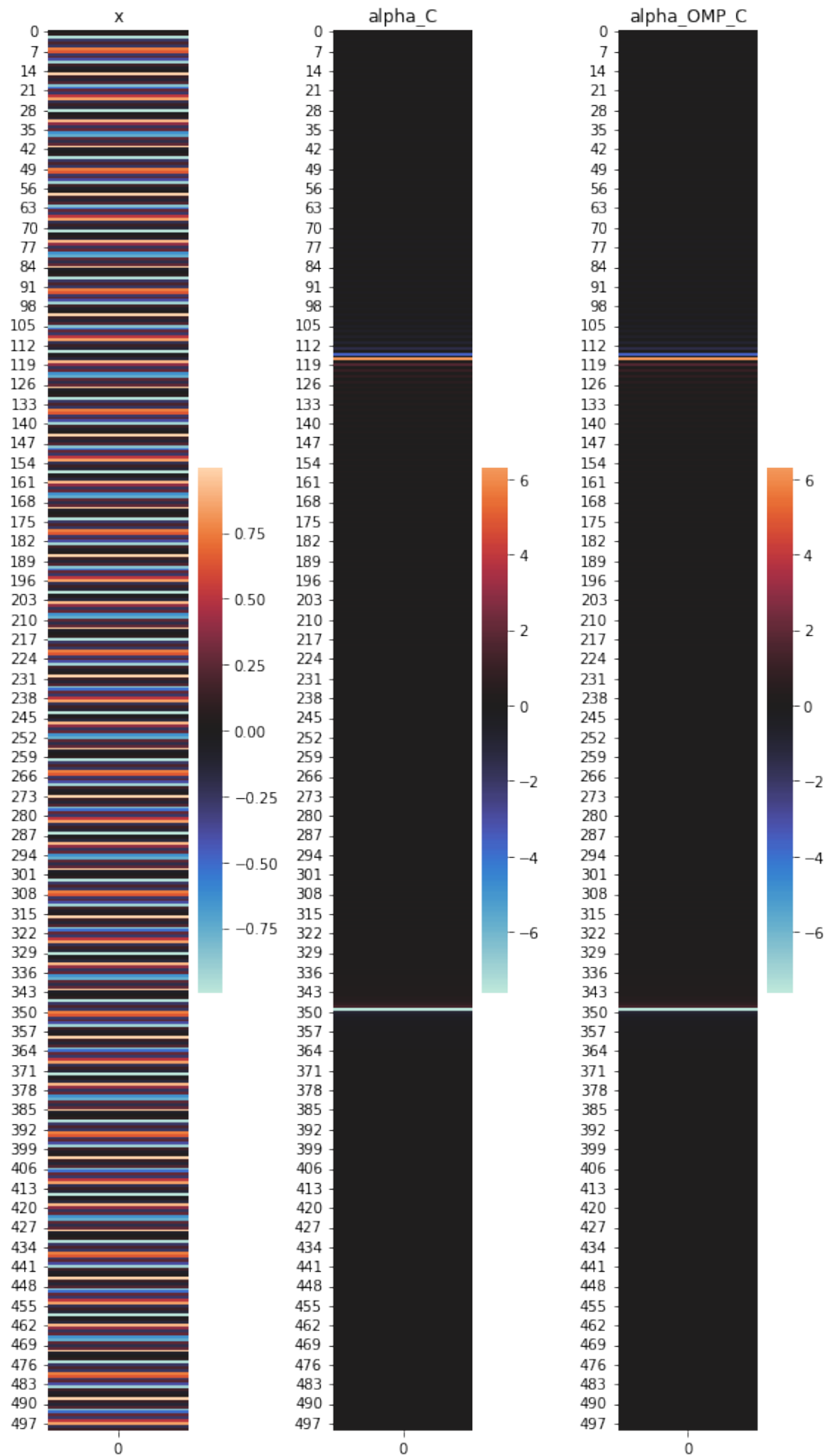


Figure 7 : Une première visualisation du signal  $x$  et ses deux représentations parcimonieuses  $\alpha_C$  et  $\alpha_{OMP_C}$



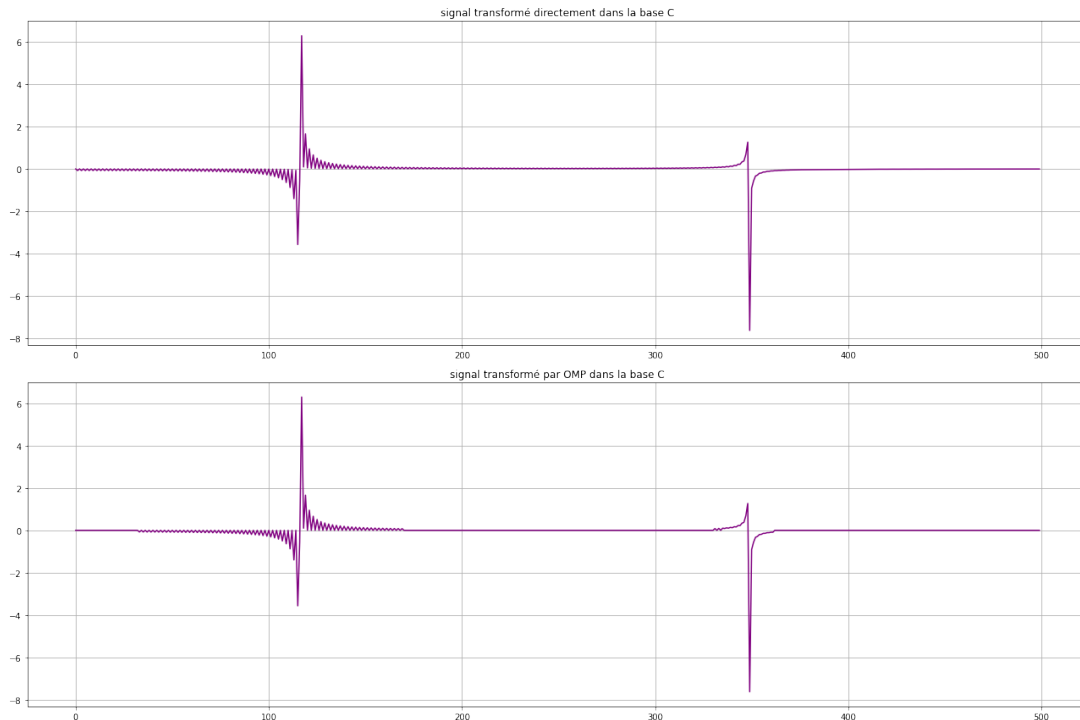


Figure 8 : Une deuxième visualisation de  $\alpha_C$  et  $\alpha_{OMP_C}$

De plus la norme de la différence entre  $\alpha_C$  et  $\alpha_{OMP_C}$  donne le résultat suivant :

$$\|\alpha_{OMP_C} - \alpha_C\|_2 = 0.627867493709721$$

Ce qui signifie qu'il y'a une toute petite différence entre les deux représentations parcimonieuses  $\alpha_C$  et  $\alpha_{OMP_C}$ .

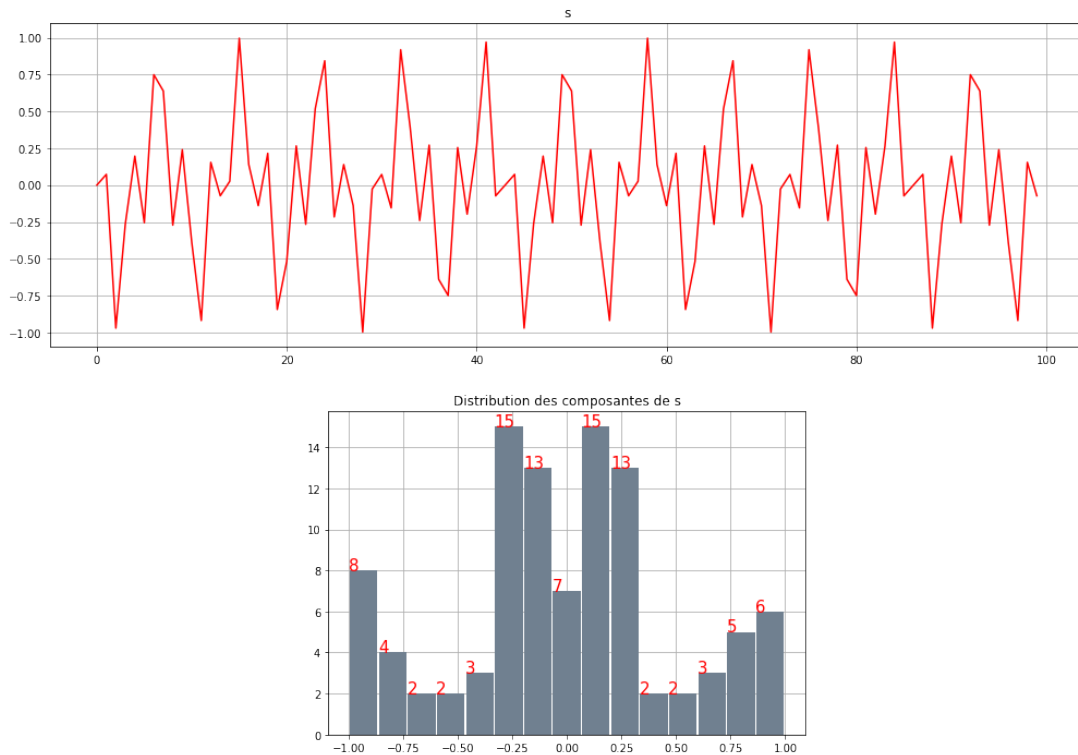
### 1.3 Représentation parcimonieuse d'un signal N=100

#### 1.3.1 Construction d'un signal

On procède de la même manière que dans la section précédente, sauf que cette fois on prend  $N = 100$ . On commence par construire un signal  $s$  tel que :

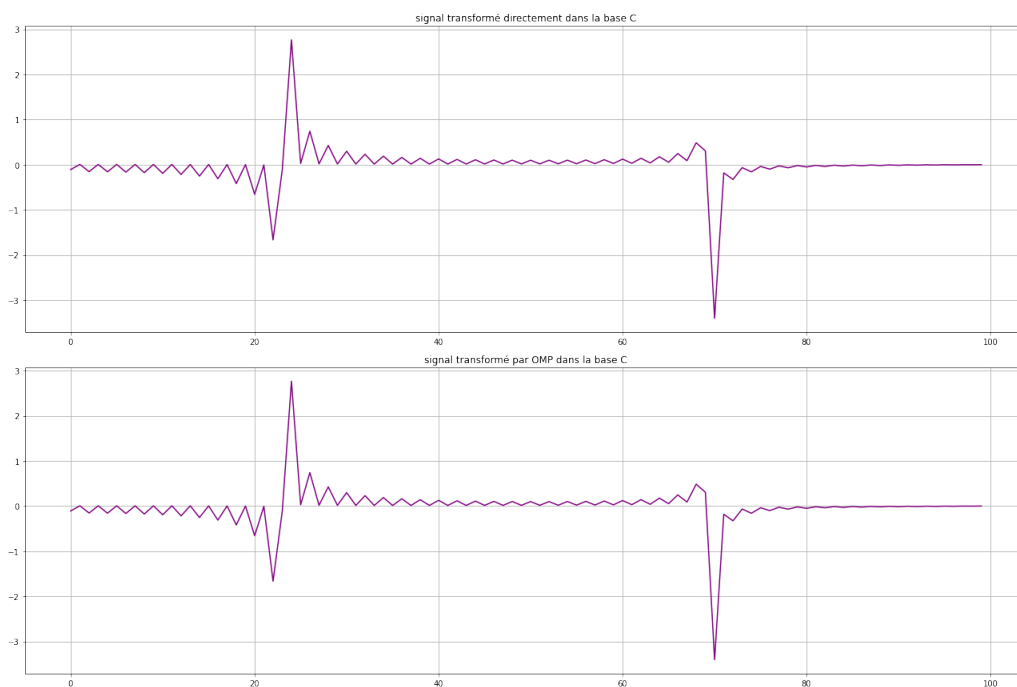
$$s(t) = \sin(2\pi f_0 t) \cos(2\pi f_p t) \quad (f_p = 100 \text{ et } f_0 = 50)$$

Ainsi, on obtient le signal visualisé ci-dessous :


 Figure 7 : Signal  $s$  et distribution de ses composantes

### 1.3.2 Représentation parcimonieuse directe et par OMP dans la base $C$ et comparaison des résultats

En calculant la représentation parcimonieuse de la même manière que dans la section précédente : directement dans la base  $C$  puis à l'aide de l'algorithme OMP dans la base  $C$ , on obtient les résultats suivants :


 Figure 8 : Une première visualisation de  $\alpha_C$  et  $\alpha_{OMP_C}$  pour  $N = 100$

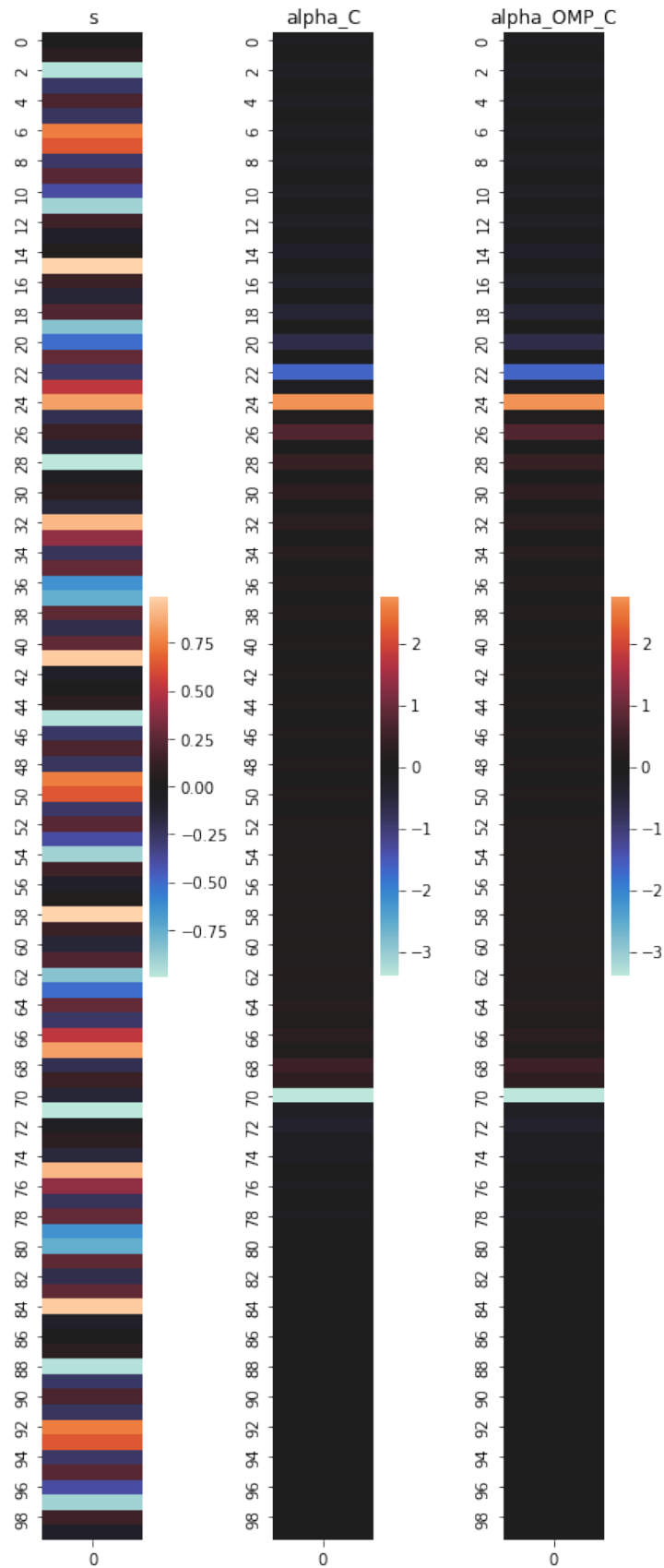


Figure 9 : Une deuxième visualisation de  $\alpha_C$  et  $\alpha_{OMP_C}$  pour  $N = 100$

Les deux représentations parcimonieuses sont très similaires, comme dans le cas précédent ( $N = 500$ ), ainsi la norme de la différence entre  $\alpha_C$  et  $\alpha_{OMP_C}$  donne le résultat suivant :

$$\|\alpha_{OMP_C} - \alpha_C\|_2 = 0.00058330762913$$

Ce qui signifie que les représentations parcimonieuses  $\alpha_C$  et  $\alpha_{OMP_C}$  sont presque les mêmes contrairement au cas précédent ( $N = 500$ ) où la visualisation montrait que les signaux étaient les mêmes mais la norme de leur différence était de l'ordre de  $10^{-1}$ .

### 1.3.3 Représentation parcimonieuse directe et par OMP dans la base F et comparaison des résultats

De la même manière, cette fois, ci-dessous la représentation parcimonieuse directe dans la base F ( $\alpha_F$ ) et la représentation parcimonieuse par l'algorithme OMP ( $\alpha_{OMP_F}$ ) :

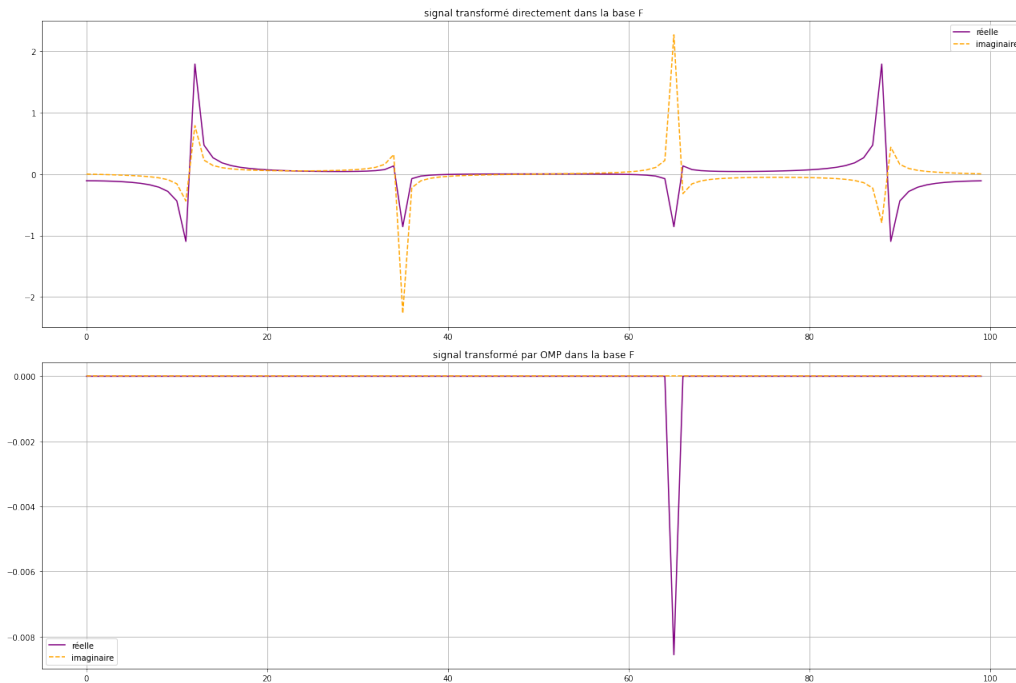


Figure 10 : Une première visualisation de  $\alpha_F$  et  $\alpha_{OMP_F}$  pour  $N = 100$

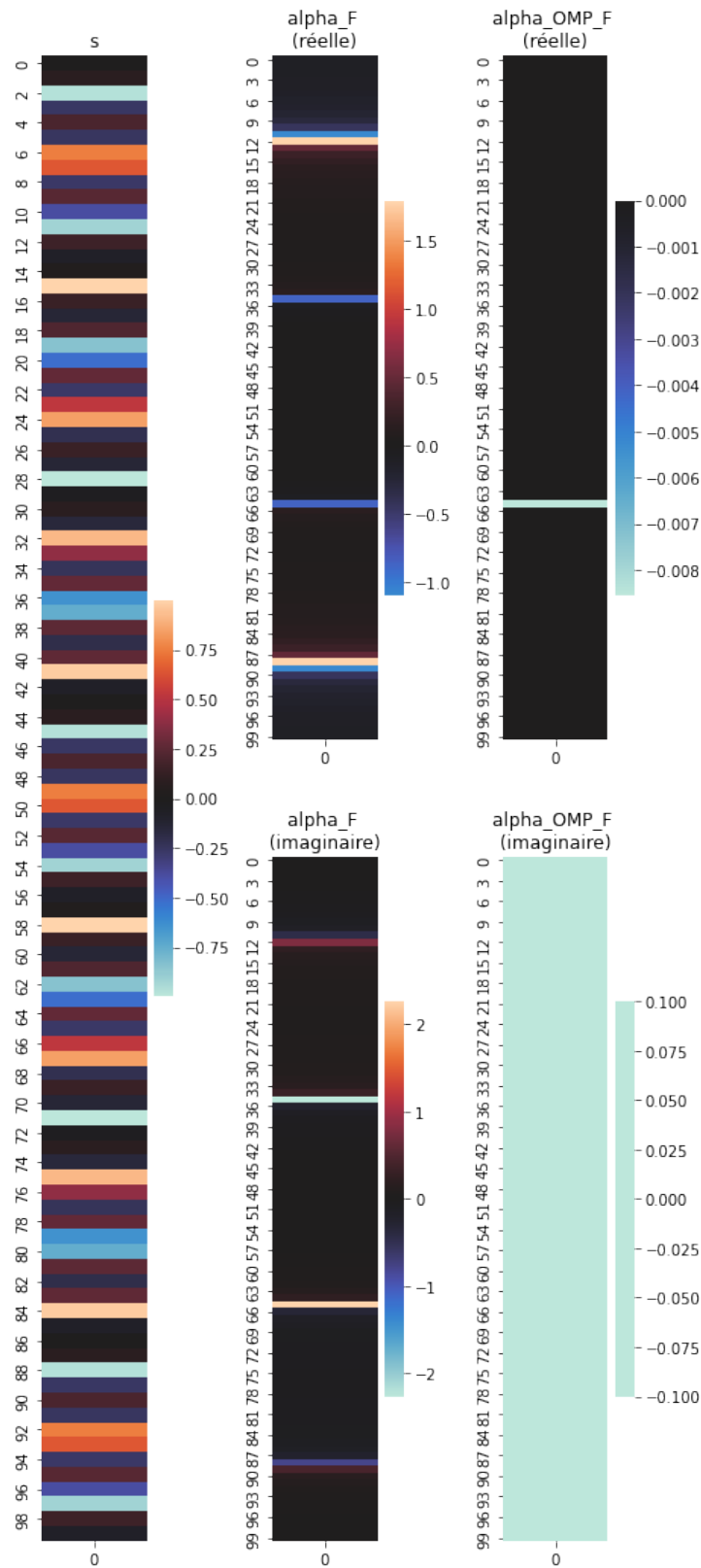


Figure 11 : Une deuxième visualisation de  $\alpha_F$  et  $\alpha_{OMP_F}$  pour  $N = 100$

Contrairement aux comparaisons entre les représentations parcimonieuses, établies dans les sections précédentes, cette fois la norme de la différence entre  $\alpha_C$  et  $\alpha_{OMP_C}$  donne un résultat plus grand :

$$\|\alpha_{OMP_F} - \alpha_F\|_2 = 4.9729777775422$$

Ceci signifie que les représentations parcimonieuses  $\alpha_F$  et  $\alpha_{OMP_F}$  sont assez différentes contrairement au cas précédent, de plus la figure ci-dessus montre que l'algorithme OMP a sélectionné une seule composante non nulle pour la représentation parcimonieuse ( $\alpha_{OMP_F}$ ) contrairement à la représentation parcimonieuse directe  $\alpha_F$  qui contient au moins 6 composantes non nulles.

## 1.4 Algorithmes de type OMP et dictionnaires formant une base

Les algorithmes de type OMP, à une itération  $k$ , risquent d'avoir un résidu colinéaire à l'un des vecteurs constituant la base ainsi pour les prochaines itérations le résidu sera orthogonal aux atomes déjà sélectionnés (qui sont des vecteurs formant la base), ce qui fait que l'algorithme OMP appliqué à une base risque de tourner en boucle en ayant que des résidus  $R_{(k)}$  colinéaires aux vecteurs de la base.

## 2 Algorithme IRLS

### 2.1 Equivalence de deux problèmes de minimisation

On a :

$$\begin{aligned}
 \min \|\alpha\|_p &= \min \|\alpha\|_p^p \\
 &= \min \sum_{i=1}^n |\alpha_i|^p \\
 &= \min \sum_{i=1}^n (|\alpha_i|^{\frac{p}{2}-1})^2 |\alpha_i|^2 \\
 &= \min \sum_{i=1}^n w_i^2 |\alpha_i|^2 \\
 &= \min \sum_{i=1}^n (w_i \alpha_i)^2 \\
 &= \min \|W\alpha\|_2^2 \\
 &= \min \|W\alpha\|_2
 \end{aligned}$$

Donc  $(\mathcal{P}_p)$  est équivalent à  $(\mathcal{P}_2)$  et les coefficients de la matrice  $W = \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_n \end{bmatrix}$

avec  $w_i = (|\alpha_i|^2)^{\frac{p}{2}-1}$

### 2.2 Solution d'un problème des moindres carrés

Notre but est de trouver  $\alpha$  tel que :

$$\alpha = \underset{\alpha}{\operatorname{argmin}} \|W(D\alpha - x)\|_2$$

On pose  $e(\alpha) = (D\alpha - x)$

Donc :

$$\alpha = \underset{\alpha}{\operatorname{argmin}} \|We(\alpha)\|_2$$

On a :

$$\|We(\alpha)\|_2 = e(\alpha)^T W^T W e(\alpha)$$

Le problème est un problème des moindres carrés ordinaires, dont l'unique solution est :

$$\alpha = (W^T W)^{-1} D^T [D(W^T W)^{-1} D^T]^{-1} x$$

Donc :

$$\alpha = Q D^T (D Q D^T)^{-1} x \quad (\text{avec } Q = (W^T W)^{-1})$$

### 2.3 Représentation parcimonieuse d'un signal par IRLS

Après implémentation de l'algorithme IRLS, on l'applique au signal  $s$  défini dans la section 1.3.1 pour avoir une représentation parcimonieuse dans le dictionnaire  $C$ , on a le résultat suivant :

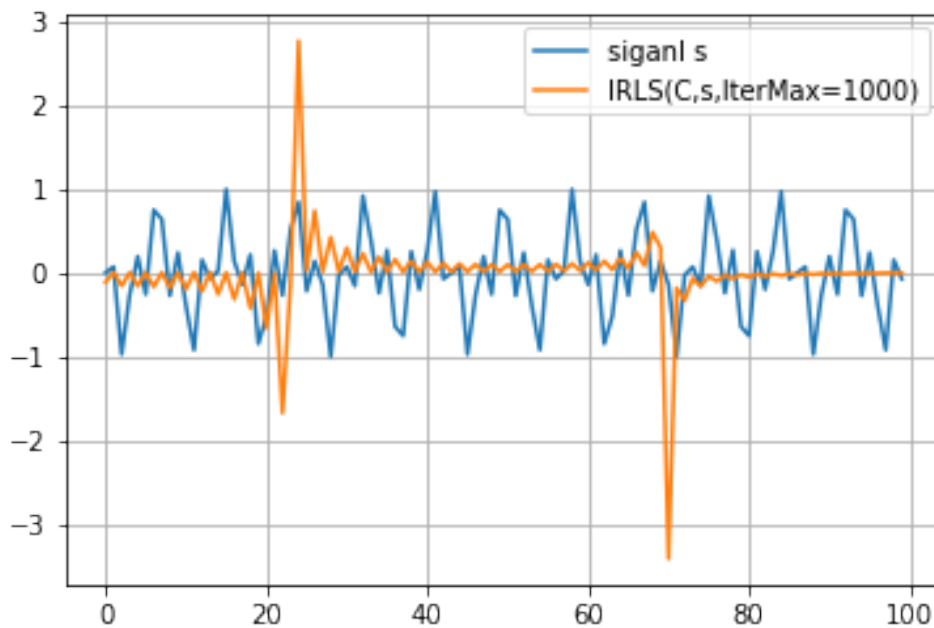


Figure 12 : Signal  $s$  et distribution de ses composantes

Dans ce cas, le résultat de la norme de la différence entre le signal original  $s$  et sa reconstruction est le suivant :

$$\|s - C\alpha_{IRLS}\|_2 = 1.313819616693 \times 10^{-13}$$

Ceci montre que l'algorithme IRLS donne une représentation parcimonieuse de bonne qualité, c'est à dire capable de faire une bonne reconstruction d'un signal donné, comme c'est le cas du signal  $s$ .

## 3 Apprentissage d'un dictionnaire (k-SVD)

### 3.1 Description des données ("DonneesProjet.xlsx")

Les données sont sous forme d'une matrice de 108 signaux utilisés ensuite pour apprendre un dictionnaire  $D_0$  à l'aide de l'algorithme k-SVD. C'est ce qu'on va voir dans la section ci-dessous.

### 3.2 Application de la k-SVD aux données

Dans cette section, on va utiliser l'algorithme k-SVD pour apprendre un dictionnaire et le rendre adapté pour les 108 signaux en terme de représentation parcimonieuse.

L'application du k-SVD à un dictionnaire  $D_0$  construit à partir des colonnes de la matrice  $X$  formée des 108 signaux, donne le résultat suivant.

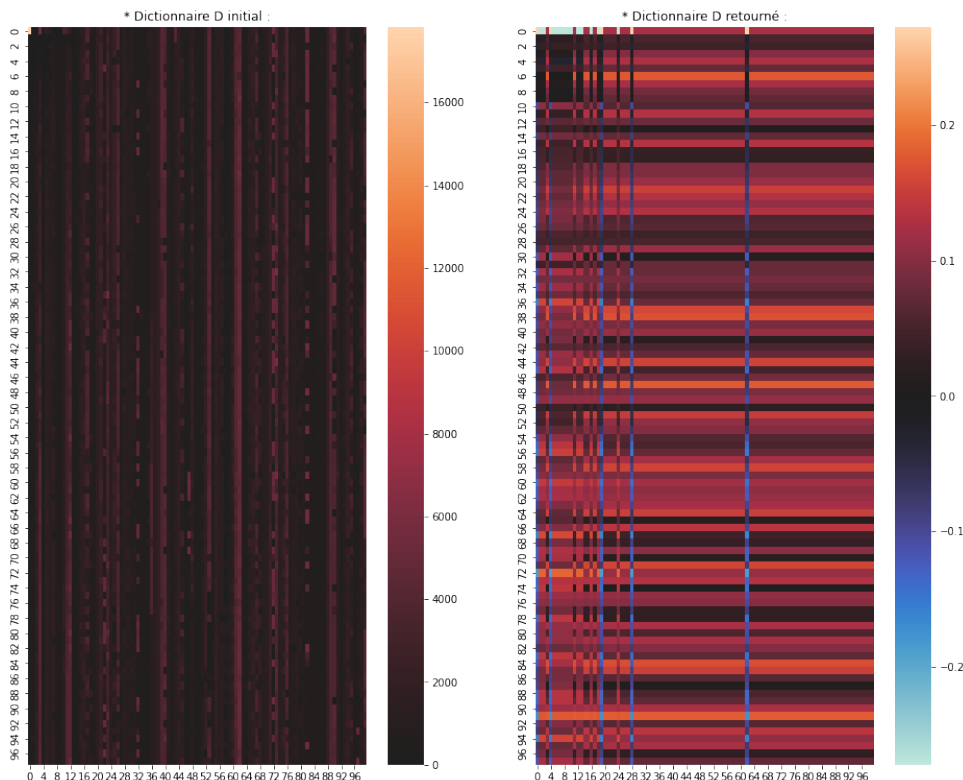


Figure 13 : Dictionnaire  $D$  avant et après application de l'algorithme d'apprentissage k-SVD

### 3.3 Comparaison des 3 Algorithmes MP, OMP et IRLS

Ci-dessous, on a construit 3 signaux test (3 colonnes pris au hasard dans la matrice  $X$  obtenu à partir de DonneesProjet.xlsx) qu'on va utiliser ensuite pour comparer les 3 algorithmes OMP, MP et IRLS.



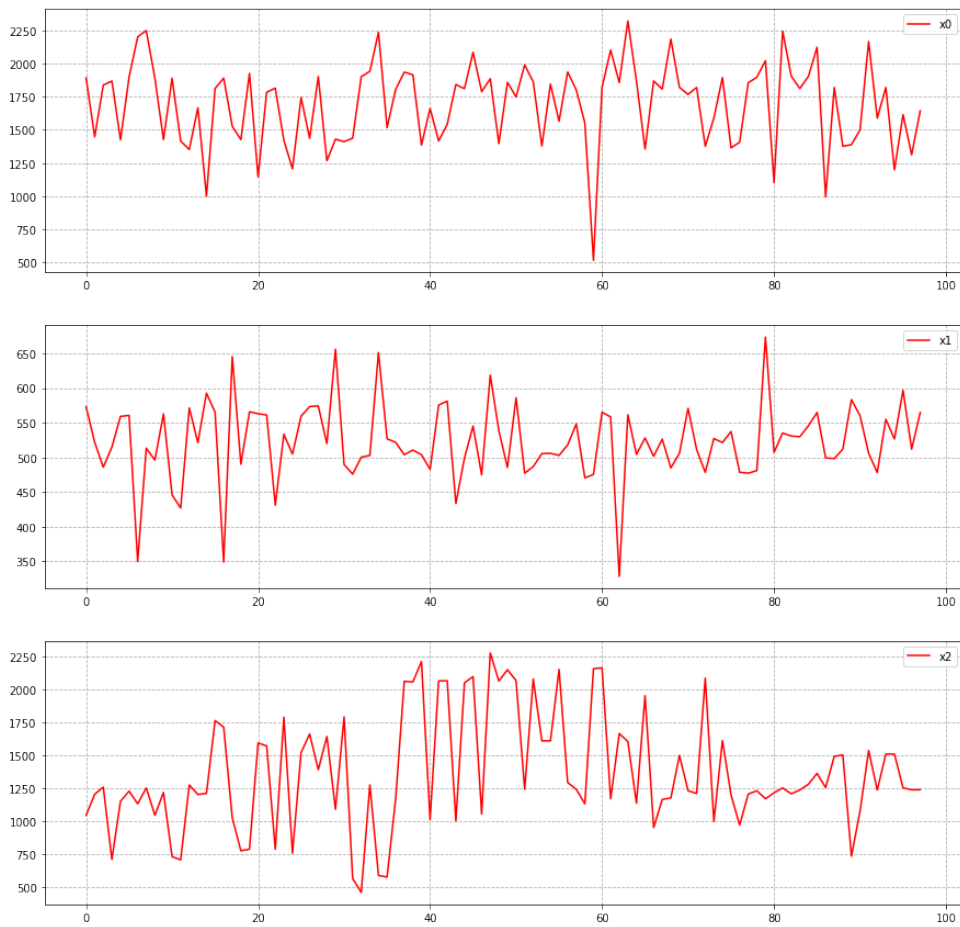


Figure 14 : Les 3 signaux test (signal  $x_0$ ,  $x_1$  et  $x_2$ )

La figure ci-dessous montre la performance de la méthode MP en terme d'efficacité lors de la reconstruction du signal test 0.

\* MP(signal\_test\_0) \*

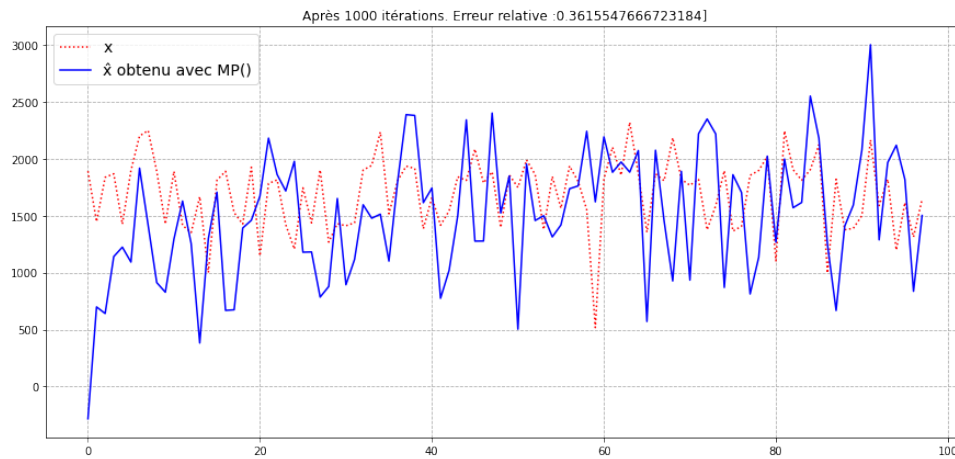
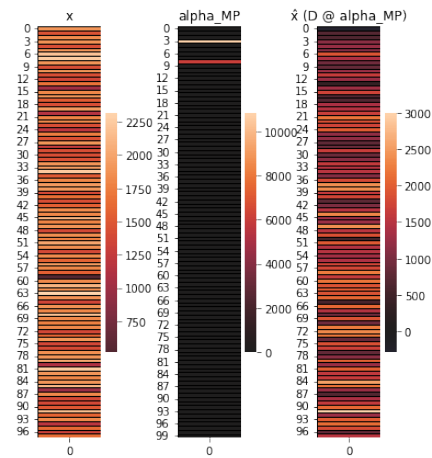


Figure 15 : Performance de la méthode MP sur le signal  $x_0$

La figure ci-dessous représente une comparaison complète des 3 algorithmes pour le signal test  $x_2$ , avec des visualisations claires et compréhensibles.

Compare results for 'signal test 2'

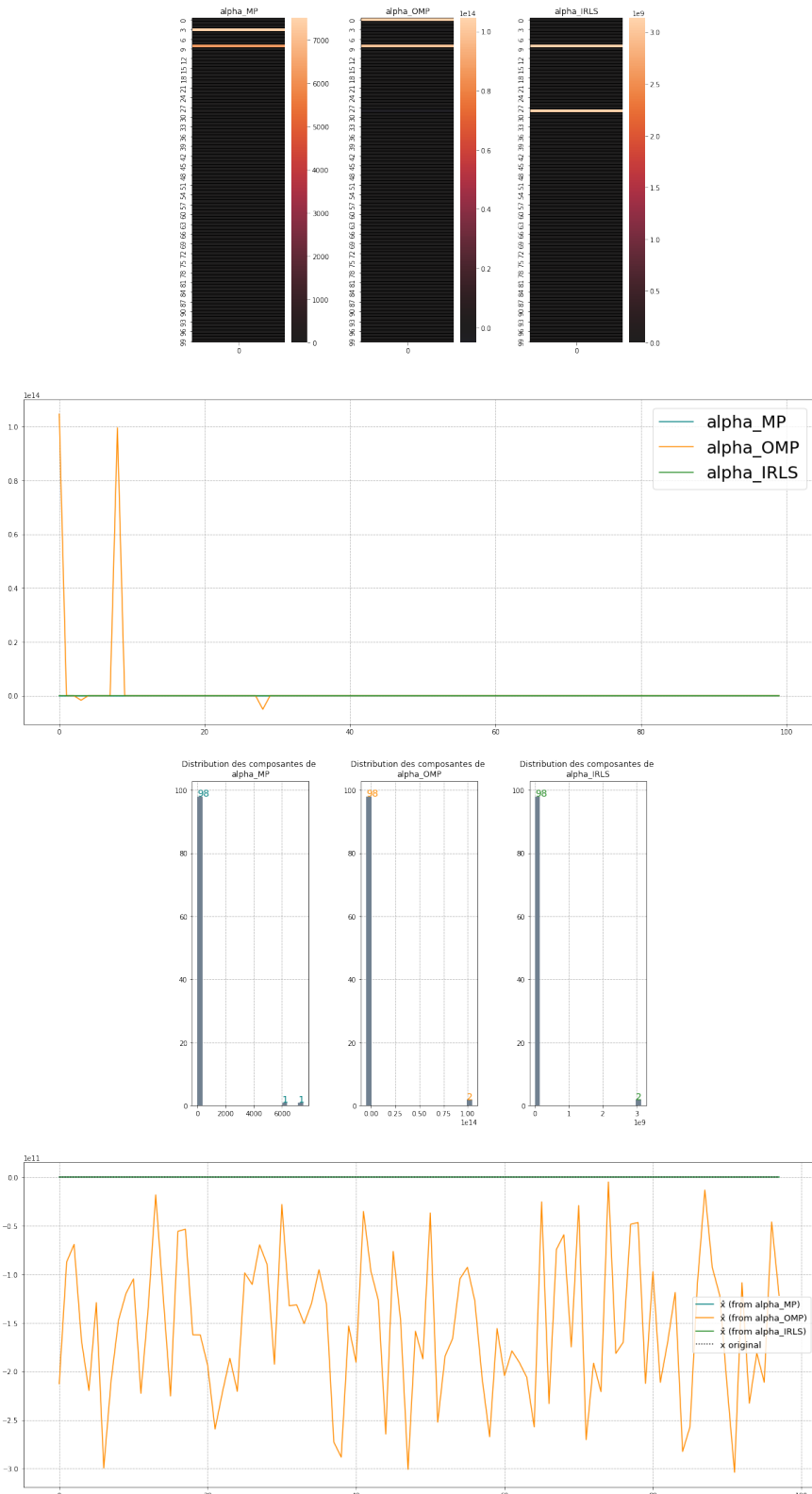


Figure 16 : Comparaison des méthodes MP, OMP et IRLS pour le signal  $x_2$

la figure ci-dessus montre le résultat du choix du meilleur algorithme ayant la meilleure performance en terme de reconstruction du signal, et ceci est pour tous les signaux  $x_0$ ,  $x_1$  et  $x_2$ . Le choix est fait en évaluant l'erreur relative entre le signal original et le signal reconstruit.

### Remarques :

Retrouver l'ensemble des résultats et graphiques obtenus, pour chaque signal, dans le notebook "compressed\_sensing\_techniques.ipynb".

```

##### * COMPARE ALPHA RESULTS : #####

-----
* signal test 0 :
-----
MP : erreur relative = 0.3615547666723184 (False)
    eval( $\alpha$ ,  $\epsilon=1e-10$ ) -> 0.97
--
OMP : erreur relative = 41227943.59796735 (False)
    eval( $\alpha$ ,  $\epsilon=1e-10$ ) -> 0.97
--
IRLS : erreur relative = 10.080806209156897 (False)
    eval( $\alpha$ ,  $\epsilon=1e-10$ ) -> 0.16
--

*** The winner is MP ! ***

-----
* signal test 1 :
-----
MP : erreur relative = 0.36620172209023677 (False)
    eval( $\alpha$ ,  $\epsilon=1e-10$ ) -> 0.97
--
OMP : erreur relative = 39589455.3248701 (False)
    eval( $\alpha$ ,  $\epsilon=1e-10$ ) -> 0.97
--
IRLS : erreur relative = 9.031837947725188 (False)
    eval( $\alpha$ ,  $\epsilon=1e-10$ ) -> 0.05
--

*** The winner is MP ! ***

-----
* signal test 2 :
-----
MP : erreur relative = 0.4218787019119257 (False)
    eval( $\alpha$ ,  $\epsilon=1e-10$ ) -> 0.97
--
OMP : erreur relative = 119553071.53950664 (False)
    eval( $\alpha$ ,  $\epsilon=1e-10$ ) -> 0.96
--
IRLS : erreur relative = 4.13420463768084 (False)
    eval( $\alpha$ ,  $\epsilon=1e-10$ ) -> 0.0
--

*** The winner is MP ! ***
  
```

Figure 17 : Sélection de l'algorithme donnant la meilleure reconstruction du signal à partir de sa représentation parcimonieuse  $\alpha$

**Remarques :**

- l'erreur relative ici est le critère pour déterminer si une reconstruction  $\hat{x}$  est exacte ou non. On déterminera qu'un signal est reconstruit parfaitement si  $\frac{\hat{x}x_2}{x_2} < 10^{-3**}$ .
- la fonction 'eval()' retourne le pourcentage de composantes nulles dans le vecteur, ainsi cela nous permet d'évaluer d'une certaine façon la parcimonie.

Au final, on observe que la représentation parcimonieuse obtenu avec l'algorithme OMP – dans le cas des 3 signaux test – est mauvaise. En effet, l'erreur relative de la reconstruction du signal est très élevé. C'est la représentation parcimonieuse obtenu avec l'algorithme MP qui se démarque des autres – dans le cas des 3 signaux test – en permettant de reconstruction le signal de manière très satisfaisante, comme en témoignent les erreurs relatives et le graphiques obtenues pour les 3 signaux test.

## 4 Matrices de mesure

Pour que le codage soit bon, il faut que la matrice d'observation  $\Phi$  "explore" toutes les colonnes du dictionnaire  $\Psi$ , ce que permet généralement une matrice aléatoire.

Dans nos TD, nous avons vu 5 matrice aléatoire  $\Phi$  différentes :

- $\Phi_1$  : matrice aléatoire générée à partir d'un processus uniformément distribué :  $U(0, 1)$
- $\Phi_2$  : matrice aléatoire générée à partir d'un processus bernoullien  $\{-1, 1\}$  :  $B(p) \rightarrow -1$  ou  $1$
- $\Phi_3$  : matrice aléatoire générée à partir d'un processus :  $B(p) \rightarrow 0$  ou  $1$
- $\Phi_4$  : matrice aléatoire générée à partir d'un processus :  $N(0; \frac{1}{M})$
- $\Phi_5$  : matrice creuse (parcimonieuse) générée de façon aléatoire

Il faut que les éléments de la matrice d'observation  $\Phi$  ne puissent pas être représentés parcimonieusement sur le dictionnaire  $\Psi$  de décomposition parcimonieuse du signal, et inversement. Ceci peut se mesurer à l'aide de la cohérence mutuelle. Entre deux matrices  $\gamma$  et  $\Phi$ , elle se note :  $\mu(\gamma, \Phi) = \sqrt{N} \cdot \max_{i,j} \frac{|\langle \gamma_{i,\bullet}, \Phi_{\bullet,j} \rangle|}{\|\gamma_{i,\bullet}\| \|\Phi_{\bullet,j}\|}$

Pour faire notre choix parmi ces matrice aléatoires. On a implémenter un programme basé sur le calcul de la cohérence mutuelle entre les matrice aléatoire et notre dictionnaire  $D$  (obtenu avec notre méthode k-svd), On affiche nos résultats sur un Graphique :

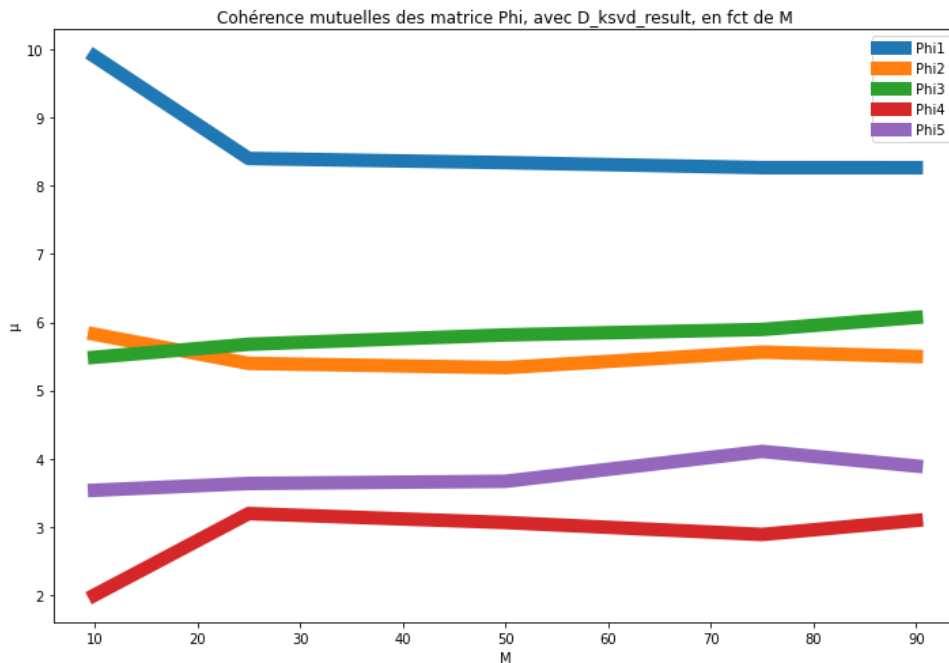


Figure 18 : Courbes de l'évolution cohérentes mutuelles en fonction de M

Parmi les 5 matrices, c'est la matrice  $\Phi_4$  qui est la moins cohérente avec la matrice D. On choisit donc la matrice  $\Phi_4$ , matrice aléatoire obtenue par un schéma de loi normale, comme matrice de mesure pour reconstruire nos signaux tests.

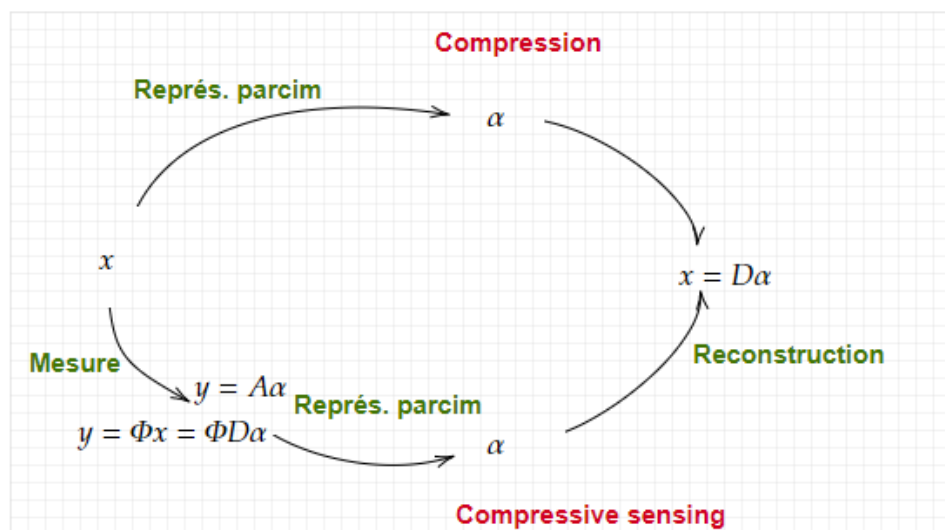


Figure 19 : Schéma - Compressing Sensing.

Pour reconstruire nos signaux tests nous allons dans un premier temps calculer  $y$  tel que  $y = \Phi x$ .

On reprendra le dictionnaire obtenu avec le k-svd. C'est avec lui qu'on calculera.

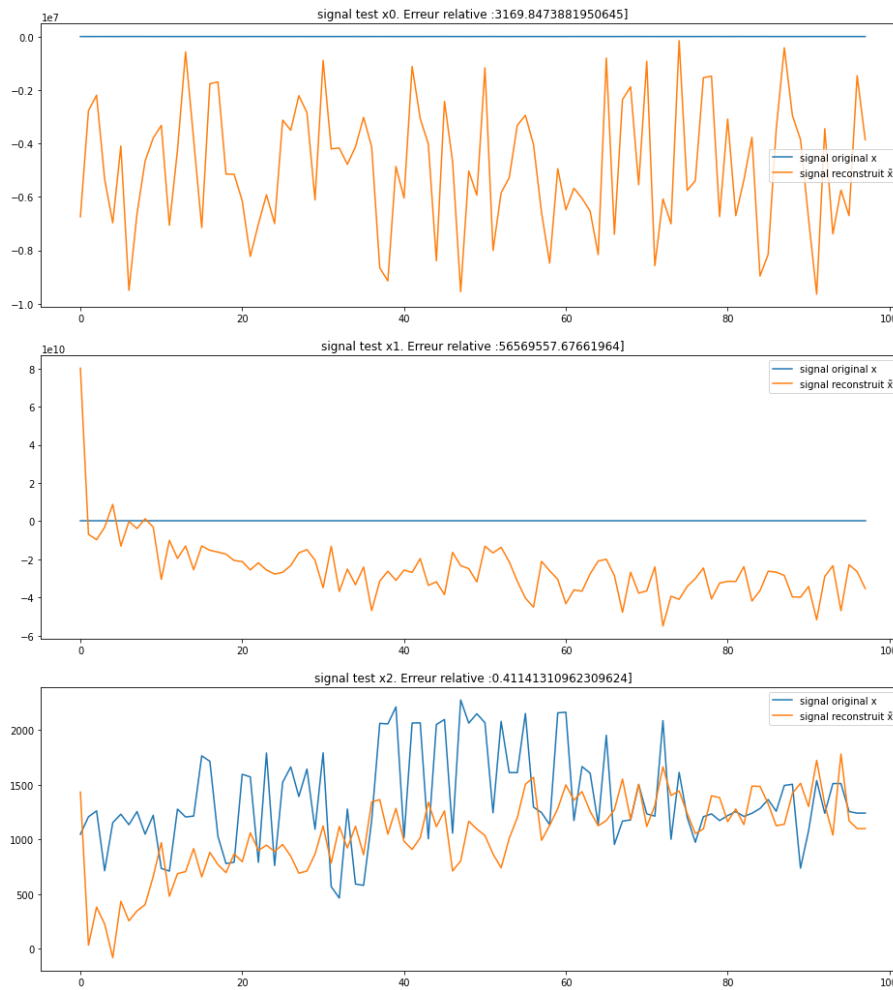


Figure 20 : Reconstruction des signaux.

## 5 Conclusion

Avec le Big Data, le flux des données massives qui ne cesse de croître, l'acquisition comprimée (Compressed Sensing) est un des grands enjeux de cette décennie.

C'est pourquoi, nous sommes très heureux d'avoir pu participer à ce projet. Dans celui-ci, nous avons pu mettre en pratique les techniques et théories vues en cours. Ce projet nous a permis de consolider nos connaissances en programmation Python, mais aussi en visualisation des données.