

TP 1 Matlab – Transformée en ondelettes (TO) sur des signaux 1D

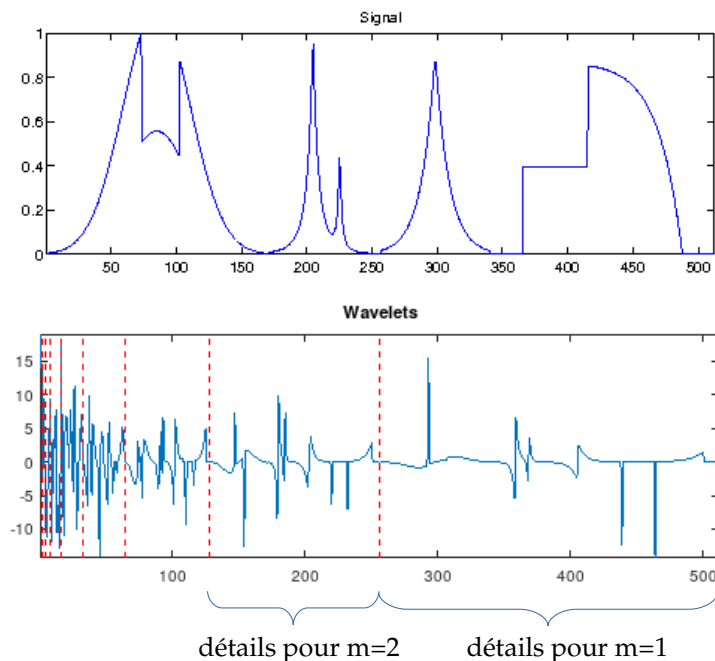
1) But du TP

Le but de ce TP est d'implanter la TO discrète dyadique à l'aide des fonctions d'ondelettes de **Haar** telle que vue en cours. Cette TO s'appliquera à des **signaux 1D discrets finis**. A l'**analyse**, on affichera les **coefficients d'ondelettes** obtenus à chacun des niveaux d'échelle, après les avoir calculés sur le maximum possible de niveaux. On passera ensuite à la **synthèse** et on affichera :

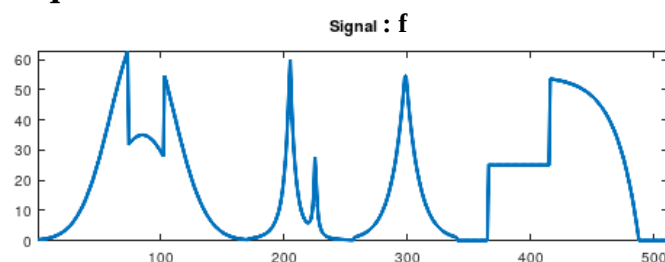
- plusieurs **reconstructions** partielles (grossières) du signal à des niveaux d'échelles intermédiaires,
- les **approximations** du signal obtenues lorsqu'on met à zéro les coefficients d'ondelettes ayant les **amplitudes** les plus faibles (après une reconstruction sur tous les niveaux) : cela donne une bonne idée du principe utilisé en **compression** de signaux.

2) Notations

On a l'habitude, après analyse en ondelettes d'un signal 1D (tel que celui illustré ci-dessous, habituellement représenté par un **vecteur colonne**), de **concaténer** tous les **détails** obtenus (cf. deuxième illustration ci-dessous appelée «**Wavelets**» et obtenue après analyse en ondelettes sur le maximum de niveaux d'échelle possible, ici 10 niveaux). On obtient ainsi un vecteur colonne contenant le même nombre d'échantillons qu'initialement (ici $512 = 2^9$ dans l'exemple).

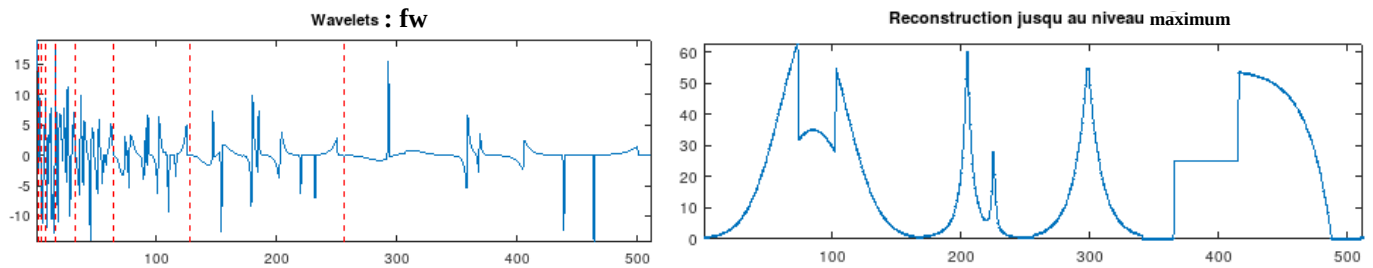


3) Illustration de ce que vous devez obtenir



3.1. Analyse de f sur le maximum de niveaux d'échelle possibles

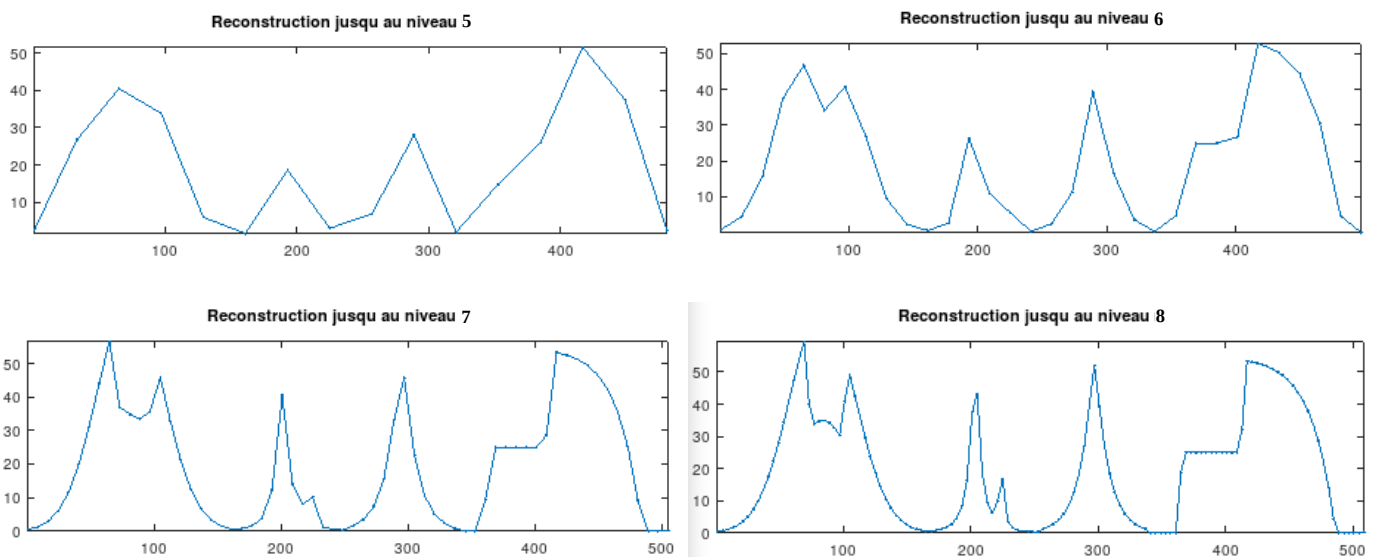
a) Analyse et synthèse sur le maximum de niveaux possibles (10 dans notre exemple car $2^9=512$).
Reconstruction parfaite



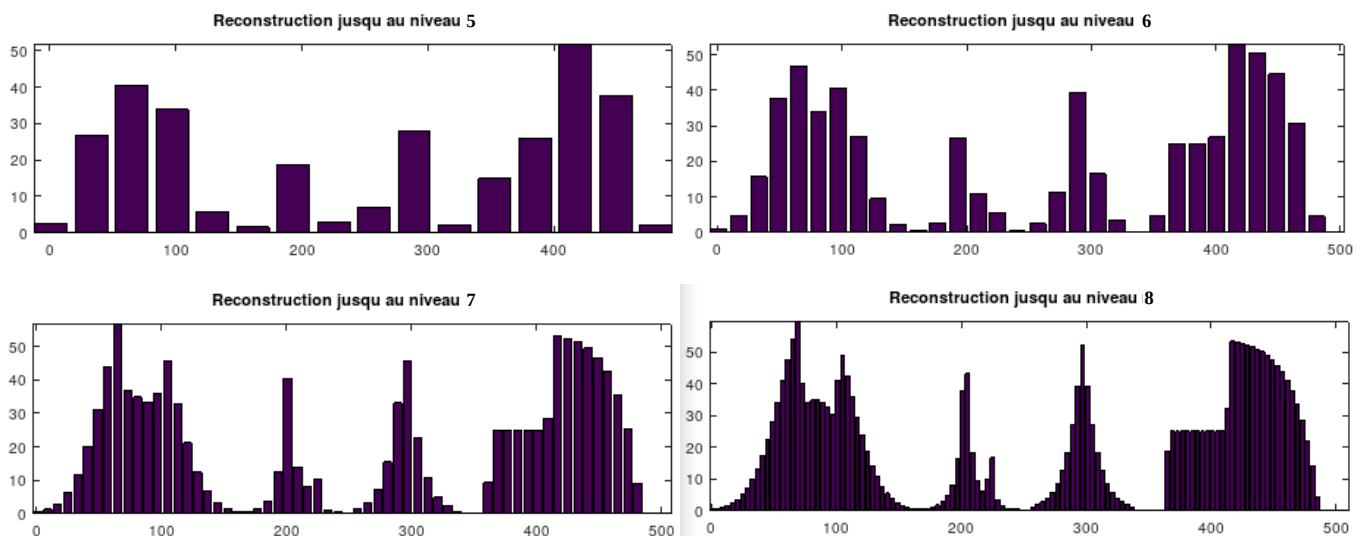
avec la fonction **plot_wavelet(fw, 0)** : fournie

b) Analyse sur le maximum de niveaux possibles (10) et synthèse sur 5, 6, 7 ou 8 niveaux seulement : reconstructions partielles

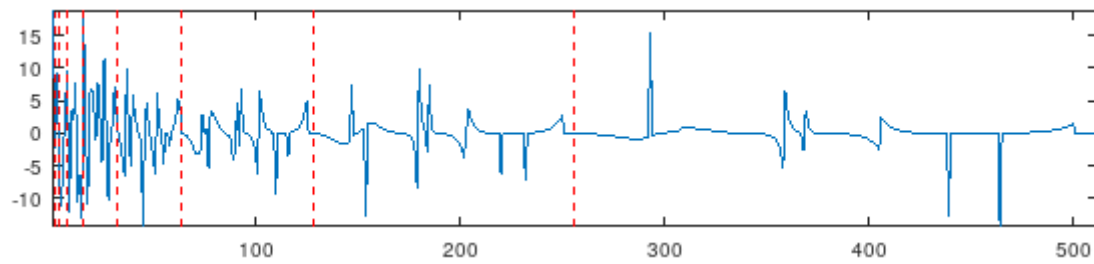
* Avec la fonction Matlab ou Octave **plot** :



* Avec la fonction Matlab ou Octave **hist** ou **bar** :

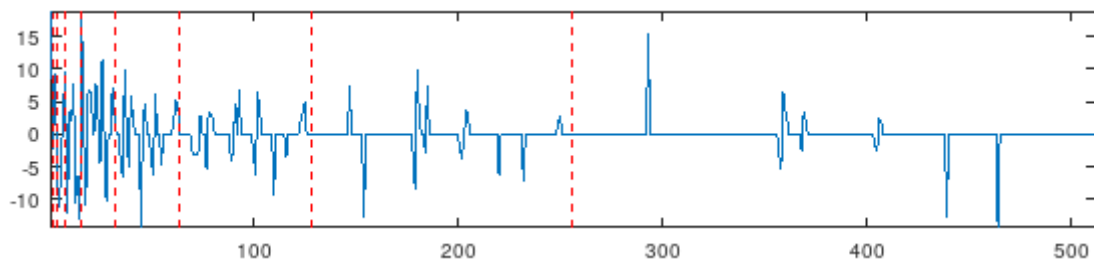


3.2. Seuillage de l'amplitude des coefficients d'ondelettes (sur le maximum possible de niveaux = 10) et reconstruction



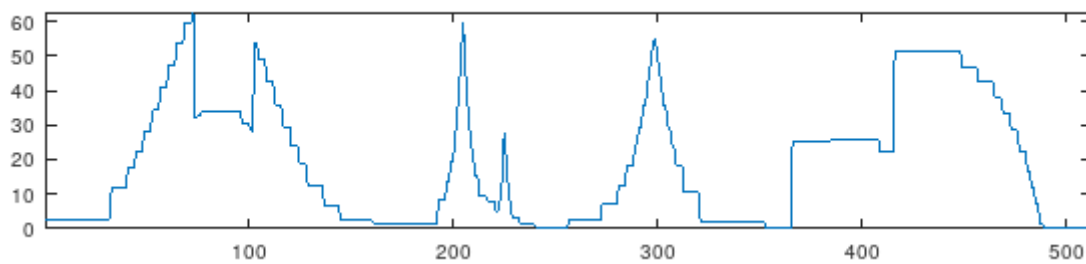
coefficients d'ondelettes non seuillés

a) A partir d'une valeur de **seuil T**

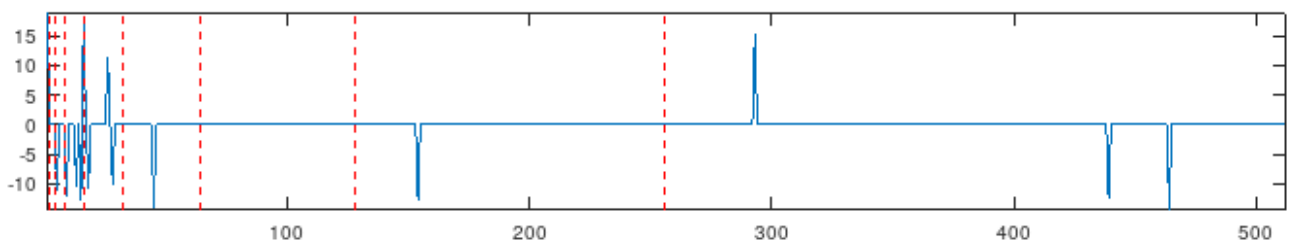


coefficients d'ondelettes seuillés (seuil $T = 2$)

T=2

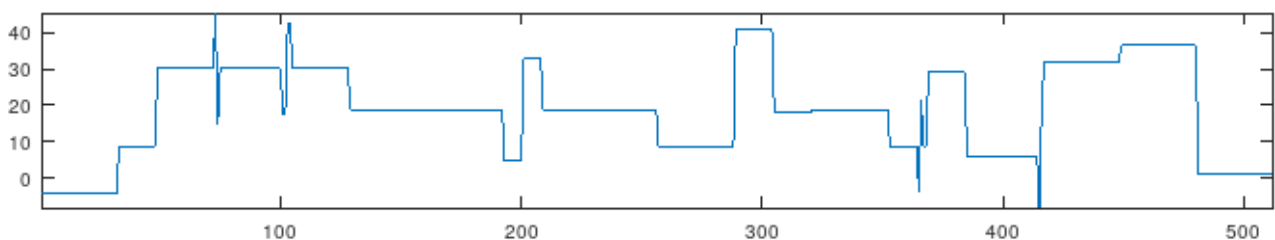


reconstruction du signal après seuillage



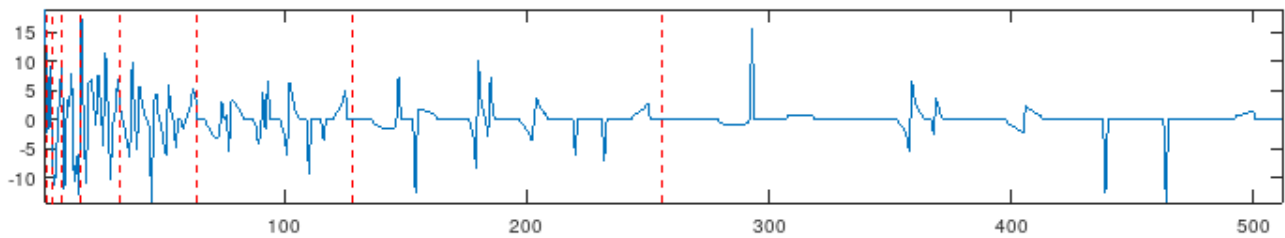
coefficients d'ondelettes seuillés (seuil $T = 10$)

T=10

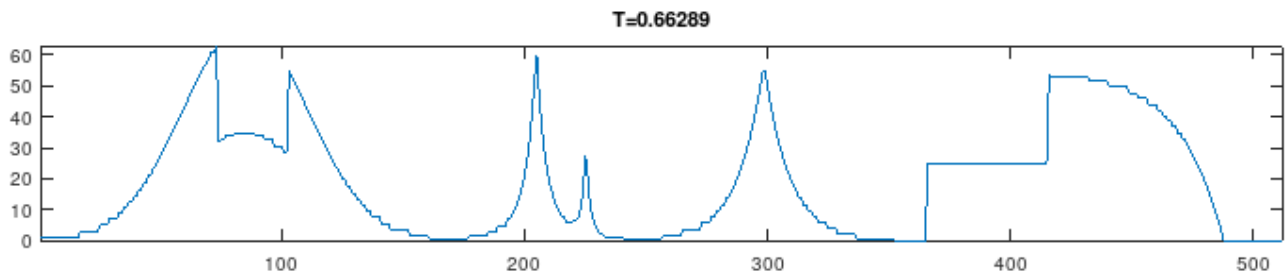


reconstruction du signal après seuillage

b) A partir d'un nombre de coefficients à conserver



conservation des 200 coefficients d'ondelettes d'amplitude maximale et mise à 0 des autres



reconstruction du signal à partir des 200 « meilleurs » coefficients (parmi les 512)

4) Fonctions/packages Matlab ou Octave à récupérer

4.1. Fonctions pour Matlab et Octave

Pour ce TP, nous utiliserons les fonctions **load_signal**, **plot_wavelet**, ...

que vous trouverez sur **Plubel**, partie Ondelettes, dossier 'CM et sujet de TD et TP' / 1D / TP1.zip et que vous placerez dans votre répertoire de travail.

4.2. Package Octave

Si vous utilisez Octave, vous aurez besoin d'installer et de lancer le **package signal** qu'on peut télécharger ici : <https://octave.sourceforge.io/signal/index.html>

Dans la fenêtre de commandes d'Octave, exécutez :

```
> pkg install signal-1.4.1.tar.gz % où signal-1.4.1.tar.gz est le nom de l'archive téléchargée
```

```
> pkg load signal % pour lancer le package que vous venez d'installer
```

Il se peut que vous ayez besoin des commandes suivantes (à exécuter dans un terminal sous Linux) :

```
> sudo apt-get install liboctave-dev octave-optim octave-specfun octave-control
```

```
> sudo apt-get install octave-signal
```

5) Analyse et synthèse en ondelettes : exemple simple

Commencer par analyser le **signal simple** vu en cours : $S_1 = [1, 1, 9, 5, 9, 3, 6, 14]$ (ou même celui encore plus simple : $S = [9, 7, 3, 5]$).

a) Analyse sur les 4 niveaux possibles de S_1

A vous maintenant d'en déduire une **fonction** d'analyse d'un signal 1D qui prendra en paramètre :

- le **signal 1D** à analyser (vecteur colonne de longueur une puissance de 2 **obligatoirement**),

- * et qui retournera le vecteur **SW** des coefficients (vecteur colonne).

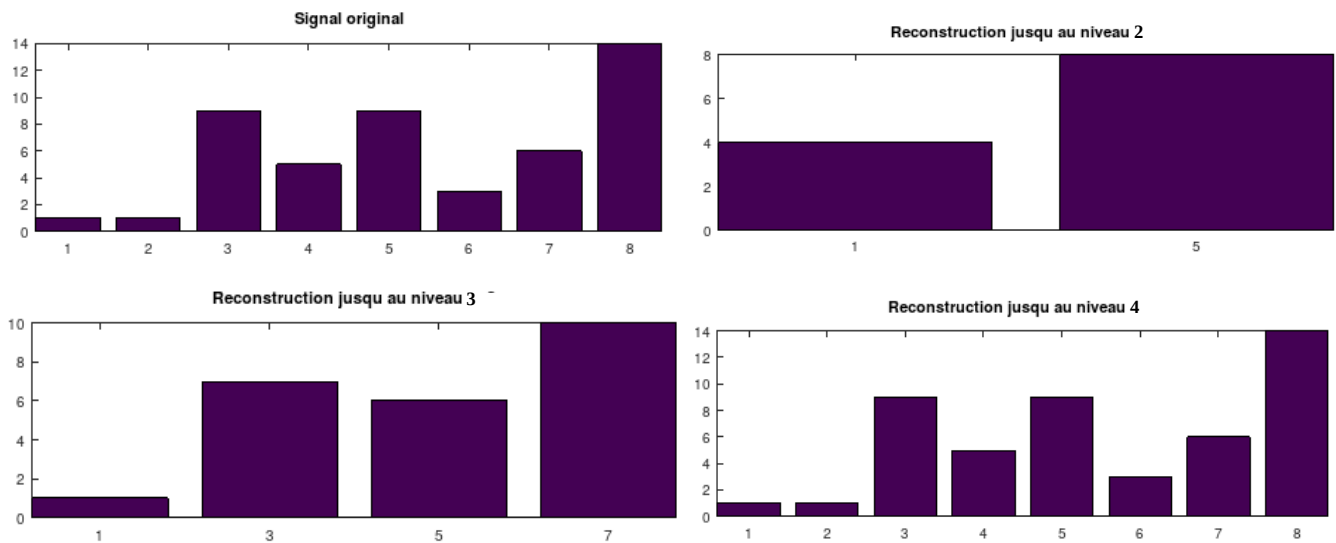
Utiliser une *boucle for* (par exemple) pour ne pas avoir à répéter les calculs à chaque niveau.

b) Synthèse et reconstructions intermédiaires pour S_1

A vous d'écrire une **fonction** qui prendra en paramètre :

- le **signal 1D** à analyser (vecteur colonne de longueur une puissance de 2 **obligatoirement**),

- le vecteur **SW** des coefficients obtenus à l'analyse (vecteur colonne de même taille que le signal),
- le nombre **nbS** de niveaux de synthèse souhaité ($0 \leq \text{nbS} \leq \text{nb max d'analyse}$),
- * qui retournera et affichera une approximation du signal après reconstruction.



Signal et 3 reconstructions possibles après analyse complète (fonction **bar** utilisée pour l'affichage)

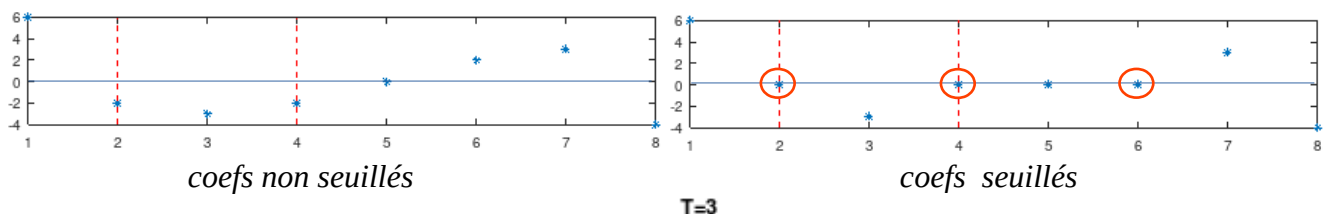
c) Seuillage de l'amplitude des coefficients d'ondelettes (sur le maximum possible de niveaux) et reconstruction

Après analyse complète (sur les 4 niv.), on obtient **SW** = [6 | -2 | -3 -2 | 0 2 3 -4]'

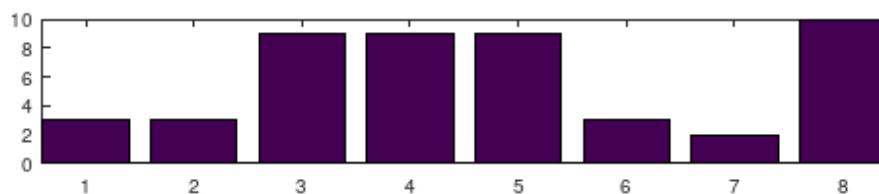
On va maintenant seuiller l'amplitude (la valeur absolue) des coefficients d'ondelettes obtenus :

$T = 3$; % définition du seuil : essayer plusieurs valeurs

SWT = **SW** .* (**abs(SW)** >= **T**); % formule pour le seuillage de l'amplitude des coefs



signal
reconstruit
après
seuillage :



6) Compléments

6.1. Signaux plus complexes

a) Signal exemple sur 512 échantillons

$N = 512$; name = 'piece-regular';

f = load_signal(name,N);

f = f-min(f); % pour que le signal n'ait que des valeurs positives

clf; subplot(2,1,1); plot(f); axis('tight'); title('Signal');

b) Autres signaux à disposition

Vous disposez des autres signaux suivants (à charger avec la fonction `load_signal`, comme pour 'piece-regular' ci-dessus) :

- 'step' (une marche d'escalier)
- 'rand' (une fonction de bruit aléatoire, différente à chaque appel)
- 'gaussiannoise' (une fonction de bruit gaussien, différente à chaque appel)
- 'regular' (une fonction lisse, différente à chaque appel)
- 'stepregular' (un mélange des fonctions step et regular)

ATTENTION tous ces signaux 1D seront représentés par des **vecteurs colonnes** !

6.2. Indications pour la fonction qui ne garde que les m coefficients d'amplitude max.

Au lieu d'indiquer un seuil T en paramètre de votre fonction, indiquez un nombre **m** pour ne garder que les **m coefficients d'amplitude (= valeur absolue) maximale** : tous les autres coefs seront mis à zéro. Utiliser pour cela la fonction `sort(abs(SW(:)), 'descend')` qui trie les coefs dans l'ordre décroissant de leurs amplitudes. Essayez avec plusieurs valeurs de **m**.

Annexe 1 : Rappel des commandes Matlab ou Octave

Concaténer 2 vecteurs colonnes :

```
A = (1:4)'
```

```
B = (5:8)'
```

```
C = [A ; B];    % C = ( 1  2  3  4  5  6  7  8)'
```