

# Décomposition des signaux en ondelettes

Lise Vaudor

21 février 2014

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Principe de la décomposition en ondelettes . . . . .	2
1.2	Mise en oeuvre sous R . . . . .	8
1.3	Quelques éléments de vocabulaire . . . . .	10
<b>2</b>	<b>Approfondissements techniques</b>	<b>12</b>
2.1	Différents types d'ondelettes . . . . .	12
2.2	Taille du signal et niveau de décomposition maximal . . . . .	14
2.3	Taille des signaux et problèmes sur les bords . . . . .	16
<b>3</b>	<b>Première utilisation possible : la décomposition de la variance ou de l'énergie</b>	<b>18</b>
3.1	Principe . . . . .	18
3.2	Exemple . . . . .	19
3.3	Mise en oeuvre sous R . . . . .	20
<b>4</b>	<b>Deuxième utilisation possible : l'analyse multi-résolution(MRA : multi-resolution analysis)</b>	<b>22</b>
4.1	Principe . . . . .	22
4.2	Exemple . . . . .	22
4.3	Mise en oeuvre sous R . . . . .	24
<b>5</b>	<b>Troisième utilisation possible : Le débruitage et/ou la compression des signaux</b>	<b>25</b>
<b>6</b>	<b>Décomposition en ondelettes à chevauchement maximal (Maximum Overlap Discrete Wavelet Transform : MODWT)</b>	<b>29</b>
6.1	Principe . . . . .	29
6.2	Mise en oeuvre sous R . . . . .	33

# Chapitre 1

## Introduction

Ce tutoriel s'adresse à un public peu familier des mathématiques, souhaitant comprendre et utiliser la décomposition des signaux en ondelettes (par exemple dans le domaine des sciences sociales, de la biologie, des sciences de l'environnement, etc.).

Je l'ai conçu de manière à illustrer et expliciter au maximum les principes qui sous-tendent cette décomposition. J'ai ainsi essayé de recourir, dans la mesure du possible, à des graphiques et des exemples plutôt qu'à des équations propres à effrayer les non-initiés... J'ai aussi fait un certain nombre de choix quant aux aspects abordés : ce tutoriel n'est donc pas exhaustif, mais vise, a minima, à répondre aux besoins suivants :

- comprendre le principe des ondelettes
- savoir interpréter les résultats de la décomposition par les ondelettes
- savoir réaliser cette décomposition à l'aide du logiciel R (package "wavelets")

Je me suis largement appuyée, pour la réalisation de ce tutoriel, sur l'ouvrage de Percival (à compléter)

### 1.1 Principe de la décomposition en ondelettes

La décomposition en ondelettes est une méthode de traitement du signal qui peut être appliquée à des séries de données temporelles ou spatiales, afin d'analyser le comportement de ce signal à plusieurs échelles.

Considérons ainsi un signal  $Z$  de longueur  $N$  :  $Z = (Z_1, Z_2, Z_3, \dots, Z_N)$ . Par souci de simplicité nous supposons dans un premier temps que  $N$  est tel qu'il existe un entier  $J_{max}$  tel que  $N = 2^{J_{max}}$  (autrement dit,  $N = 2$ , ou 4, ou 8, ou 16, ou 32, ou 64, etc.).

Considérons un exemple simple, avec  $N = 16$  ( $N = 2^4$ ), cf figure 1.a.

On peut décrire  $Z$  à diverses échelles, par exemple en calculant des moyennes des "régions" de longueur 1, 2, 4, 8, et 16 points (cf figure 1.b).

Décrire ce signal à ces différentes échelles comme nous l'avons fait, à l'aide de représentations en marches d'escalier, revient en fait à le décomposer en **ondelettes de Haar** (i.e. un type d'ondelettes de forme très simple).

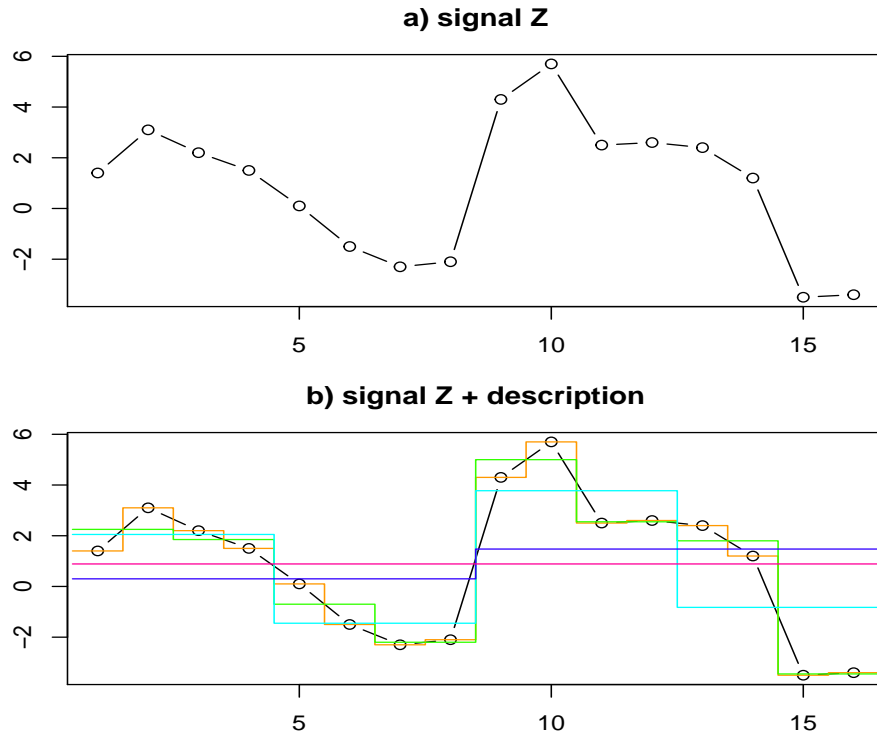


FIGURE 1 – Un exemple de signal simple  $Z$  : on peut pour décrire ce signal à différentes échelles considérer les moyennes sur l'ensemble des 16 points (en rose), sur des régions de 8 points (en violet), de 4 points (en bleu), de 2 points (en vert) et d'un seul point (en orange)

Intéressons-nous à ce qui se passe à l'échelle la plus fine (i.e. au niveau 1). Si l'on se focalise sur une portion du signal correspondant à deux points, alors on peut le décrire par deux valeurs (cf fig. 2) :

- celle qui correspond à la moyenne des deux points (matérialisée par le segment horizontal en vert)
- celle qui correspond à la différence entre les deux points (matérialisée par la flèche orange)

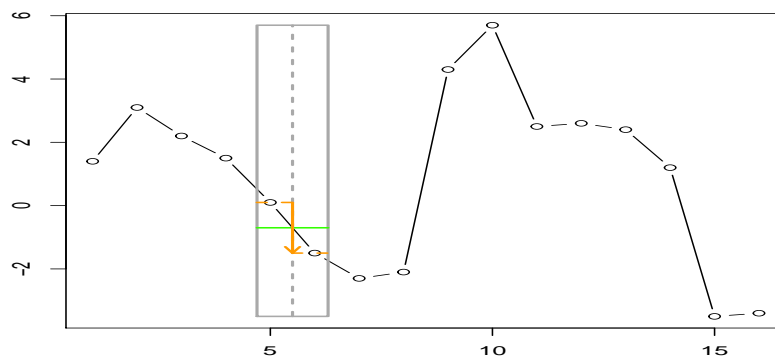


FIGURE 2 – Description d'une portion du signal à l'échelle la plus fine (niveau 1)

On peut en fait décrire l'ensemble du signal à l'échelle la plus fine de cette manière (cf fig.3). Ainsi pour

un signal de longueur  $N=16$ , on obtient 8 valeurs correspondant aux moyennes locales (en vert) et 8 valeurs correspondant aux variations locales (en orange).

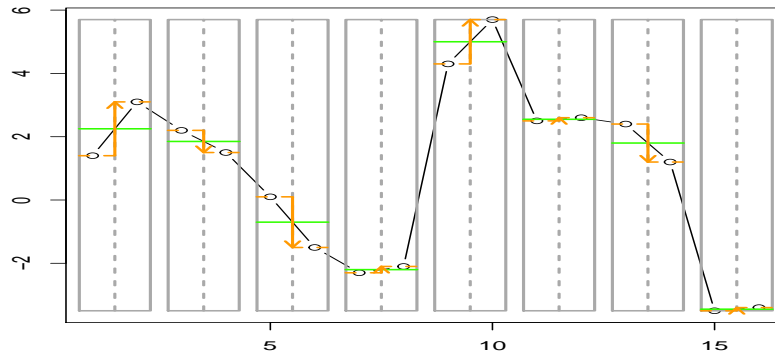


FIGURE 3 – Description du signal à l'échelle la plus fine (niveau 1)

En somme, pour décrire le processus de manière un peu imagée, tout se passe comme si on faisait glisser une "fenêtre" sur le signal, et que l'on décrivait à chaque emplacement ce qu'on y observe.

De la même manière on peut se placer à une échelle plus large (cf fig. 4). Par exemple au niveau 2, on peut décrire les portions de signal à travers des moyennes locales sur 4 points (en bleu) et des variations locales entre 2 portions de 2 points (en vert).

Ainsi, plus le niveau de décomposition est élevé, plus la "fenêtre" à travers laquelle on décrit le signal est large.

Les valeurs de **moyennes locales** et de **variations locales** sont (dans le cas des ondelettes de Haar) directement proportionnelles à ce qu'on appelle les **coefficients d'échelle** et les **coefficients d'ondelettes**. Pour les autres types d'ondelette, le principe reste similaire : les coefficients d'échelle mesurent la moyenne locale (moyenne non arithmétique) et que les coefficients d'échelle mesurent la variabilité locale.

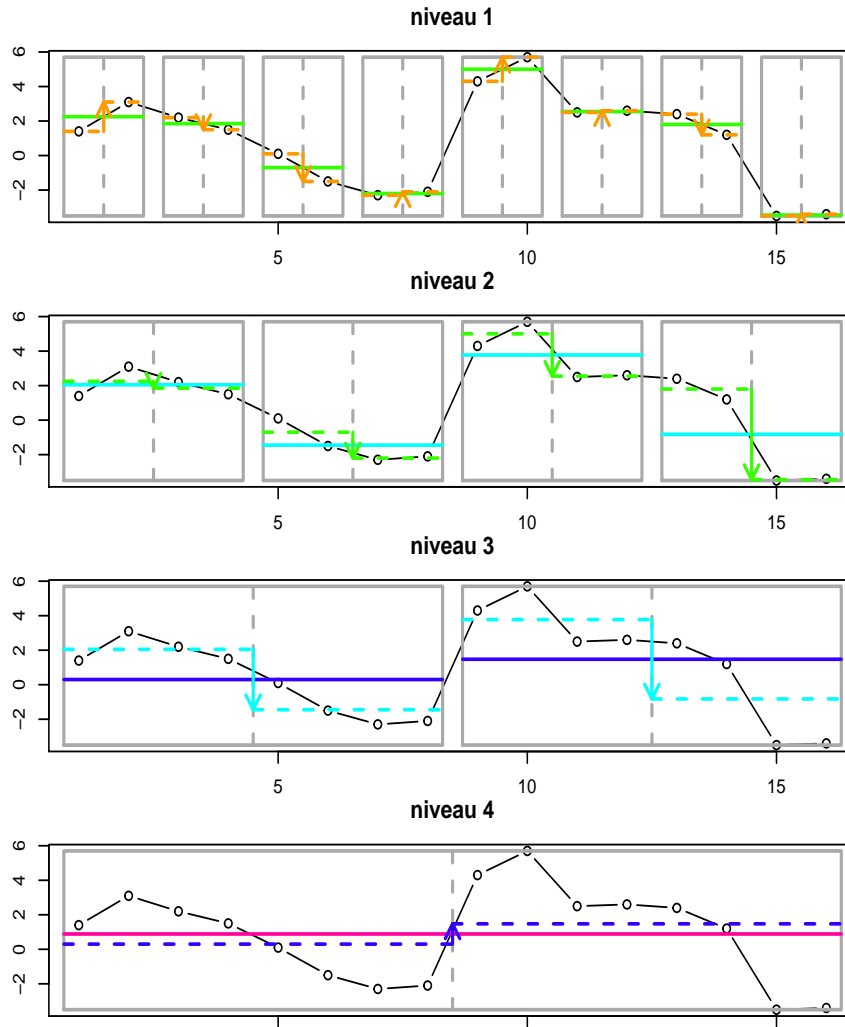


FIGURE 4 – Description du signal à l’ensemble des échelles

Enfin, considérons la figure 5. Tout à gauche, on retrouve la description du signal  $Z$  sous forme de ”moyennes régionales” à différentes échelles. Les différents niveaux de décomposition correspondent à différentes tailles de régions (donc à différentes échelles). Les coefficients d’échelle  $V$  correspondent à des **moyennes locales**, tandis que les coefficients d’ondelettes  $W$  correspondent à des **variations locales**.

Ainsi, si l’on considère que 2 points consécutifs sont écartés de  $\Delta_t$  (par exemple 1h dans le cas de séries temporelles, ou 100m dans le cas de séries spatiales), alors les niveaux de décomposition correspondent à ces échelles :

- **niveau 1=échelle  $\Delta_t$** ,
  - $W_1$  est proportionnel aux variations entre deux segments de largeur  $\Delta_t$
  - $V_1$  est proportionnel aux moyennes sur deux segments de largeur  $\Delta t$
- **niveau 2=échelle  $2\Delta_t$** ,
  - $W_2$  est proportionnel aux variations entre deux segments de largeur  $2\Delta_t$
  - $V_2$  est proportionnel aux moyennes sur deux segments de largeur  $2\Delta t$
- **niveau 3=échelle  $4\Delta_t$** ,

- $W_3$  est proportionnel aux variations entre deux segments de largeur  $4\Delta_t$
- $V_3$  est proportionnel aux moyennes sur deux segments de largeur  $4\Delta t$
- **niveau 4=échelle  $8\Delta_t$ ,**
- $W_4$  est proportionnel aux variations entre deux segments de largeur  $8\Delta_t$
- $V_4$  correspond à la moyenne sur deux segments de largeur  $8\Delta t$

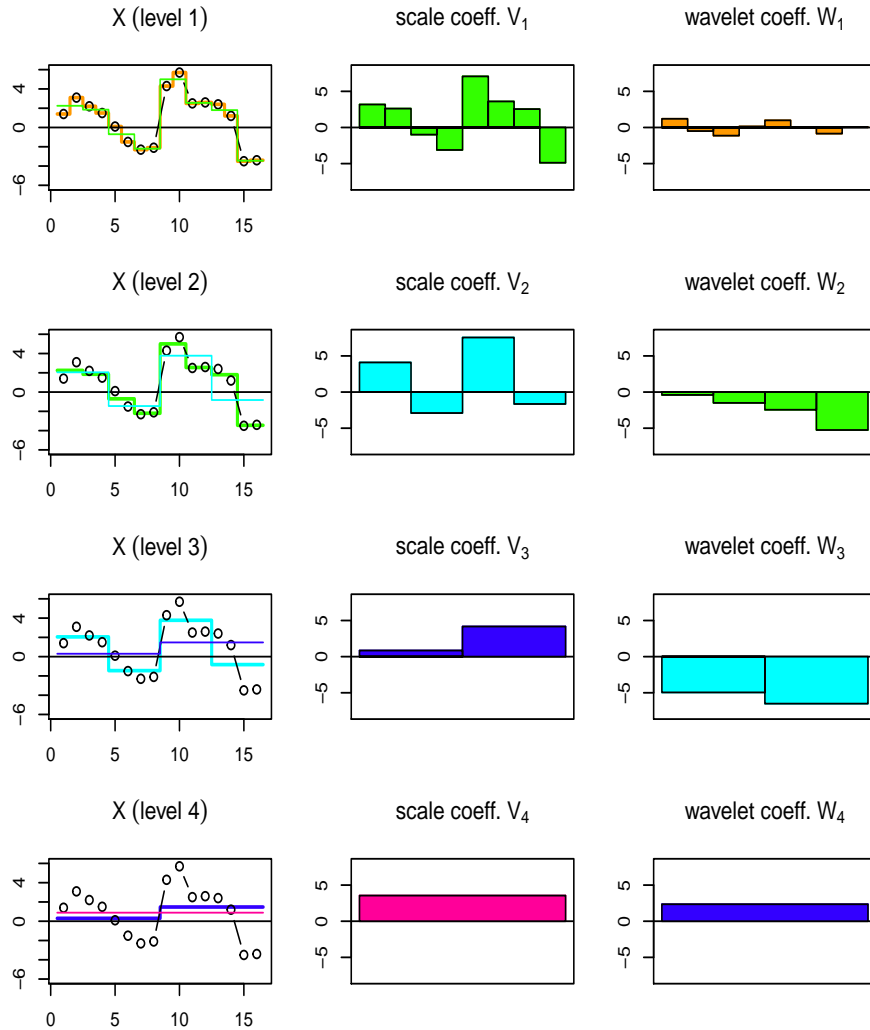


FIGURE 5 – Décomposition en ondelettes d’un signal simple  $Z$  : les différentes graphiques correspondent aux différents niveaux de décomposition. A gauche, on a représenté le signal brut (ronds noirs) ainsi que les moyennes locales propres à une échelle donnée.

On constate ainsi que pour décrire  $Z$  au niveau  $j$  (i.e. sur une échelle de  $j\Delta t$ ), on utilise des coefficients  $V_j$  et  $W_j$  de longueur  $N/(2^j)$ .



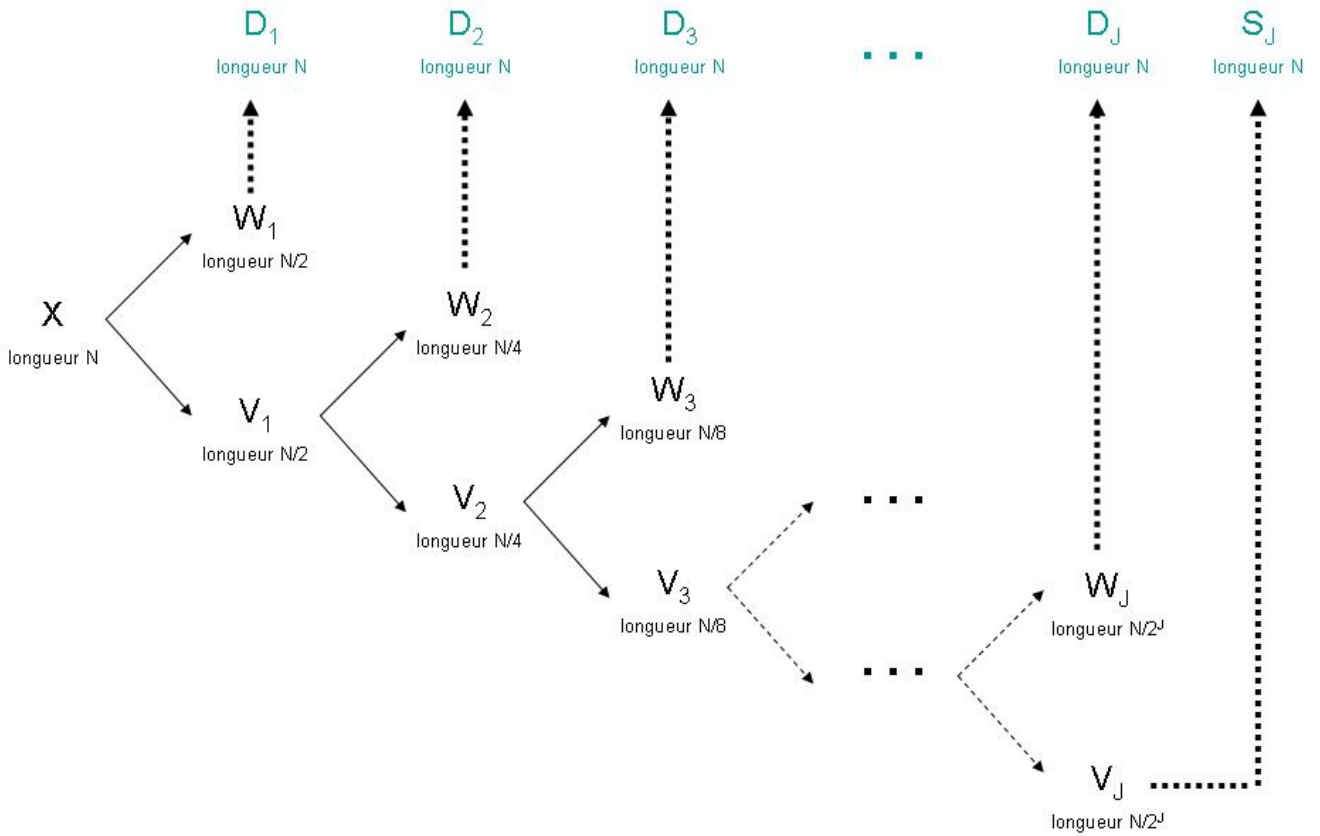


FIGURE 6 – Principe de la décomposition en ondelettes : algorithme pyramidal.

Ainsi, si on décompose autant que possible  $Z$  (c'est à dire jusqu'au niveau  $J_{max}$  tel que  $N = 2^{J_{max}}$ , on obtient  $N$  coefficients, dont

- $N/2$  correspondent au premier niveau (coefficients  $W_1$ )
- $N/4$  correspondent au deuxième niveau (coefficients  $W_2$ )
- $N/8$  correspondent au troisième niveau (coefficients  $W_3$ )
- ...
- $N/(2^J) = 1$  correspond au dernier niveau (coefficients  $V_{J_{max}}$  dans notre exemple)

Il est en fait tout-à-fait possible, mathématiquement, de déduire

- $W_2$  et  $V_2$  de  $V_1$
- $W_3$  et  $V_3$  de  $V_2$
- $W_4$  et  $V_4$  de  $V_3$

Il s'agit en fait de traiter, à chaque nouveau niveau de décomposition, les coefficients d'échelle (donc les moyennes locales) comme on a traité le signal de départ.

C'est la raison pour laquelle on parle d'algorithme pyramidal (cf figure 6, où en l'occurrence la pyramide est "couchée sur le côté"... Nous reviendrons à cette figure un peu plus tard pour expliquer ce que sont les éléments en vert).

## 1.2 Mise en oeuvre sous R

Nous allons travailler tout au long de ce tutorial avec le package "wavelets". Pour l'installer et le charger, vous pouvez procéder comme suit :

```
> install.packages("wavelets")
> require(wavelets)
```

En anglais, on parle de "Discrete Wavelet Transform", d'où le nom de la fonction qui calcule cette transformation sous R :

```
> mydwt=dwt(z,filter="haar",n.levels=4)
> slotNames(mydwt) # affiche les différents éléments renseignés dans l'objet mydwt
```

```
[1] "W"          "V"          "filter"     "level"      "n.boundary"
[6] "boundary"   "series"     "class.X"    "attr.X"     "aligned"
[11] "coe"
```

```
> # Pour afficher les coeffs d'ondelettes      > # Pour afficher les coeffs d'échelle
> print(mydwt@W)                               > print(mydwt@V)
```

```
$W1
      [,1]
[1,] 1.20208153
[2,] -0.49497475
[3,] -1.13137085
[4,] 0.14142136
[5,] 0.98994949
[6,] 0.07071068
[7,] -0.84852814
[8,] 0.07071068

$W2
      [,1]
[1,] -0.40
[2,] -1.50
[3,] -2.45
[4,] -5.25

$W3
      [,1]
[1,] -4.949747
[2,] -6.505382

$W4
      [,1]
[1,] 2.35

$V1
      [,1]
[1,] 3.1819805
[2,] 2.6162951
[3,] -0.9899495
[4,] -3.1112698
[5,] 7.0710678
[6,] 3.6062446
[7,] 2.5455844
[8,] -4.8790368

$V2
      [,1]
[1,] 4.10
[2,] -2.90
[3,] 7.55
[4,] -1.65

$V3
      [,1]
[1,] 0.8485281
[2,] 4.1719300

$V4
      [,1]
[1,] 3.55
```

On peut ainsi avoir accès aux coefficients pour un niveau avec les commandes suivantes :

```
> # Pour le niveau 3 des coeffs d'ondelettes > # Pour le niveau 3 des coeffs d'échelle
> mydwt@W$W3                               > mydwt@V$V3

           [,1]                               [,1]
[1,] -4.949747                               [1,] 0.8485281
[2,] -6.505382                               [2,] 4.1719300
```

Les coefficients d'ondelettes étant proportionnels aux variations du signal à différentes échelles, on est évidemment tenté de faire correspondre les différents coefficients aux segments du signal qu'ils décrivent.

C'est, de fait, ce qu'on a tenté de faire à des fins pédagogiques dans la figure 5.

Cependant, pour ce faire, il y a plusieurs difficultés à surmonter.

- La première est que pour un signal  $Z$  de longueur  $N$ , les coefficients  $W_1$  seront de longueur  $N/2$ , les coefficients  $W_2$  de longueur  $N/4$ , etc. Il est par conséquent difficile de faire correspondre en "face à face" les coefficients d'ondelettes des différents niveaux et le signal  $Z$  (dans la figure 5 on a joué sur la largeur des barres représentant les coefficients pour y arriver)
- La deuxième difficulté est que les coefficients sont proportionnels, mais non égaux, aux changements de  $Z$  aux diverses échelles (dans la figure 5, on n'a pas réglé ce problème).

En somme, la transformation de  $Z$  en coefficients nous fournit bien une information sur  $Z$  décomposée sur plusieurs échelles. Mais cette information a l'inconvénient de ne pas se trouver dans la même gamme de valeurs (en  $x$  comme en  $y$ ) que le signal.

Pour remédier à ces problèmes, on effectue la "transformation inverse" des coefficients d'ondelettes pour obtenir le détail de ce qui se passe à une échelle donnée (par exemple  $j$ ) pour  $Z$  : il s'agit d'un nouveau signal qu'on appelle **détail de niveau  $j$  :  $D_j$** . On effectue une transformation similaire des coefficients d'échelle pour obtenir le **signal "lissé" de niveau  $j$ , (ou "smooth" en anglais),  $S_j$** .

$D_j$  et  $S_j$  sont donc calculés à partir de  $W_j$  et  $V_j$  respectivement, et sont tous deux de taille  $N$  (cf figure 6)

Ainsi, pour résumer cette "chaîne" de traitements :

- On s'intéresse aux moyennes et variations de  $Z$  à une échelle donnée  $j$ .
- On effectue le calcul des coefficients d'échelle  $V_j$  et des coefficients d'ondelette  $W_j$
- On transforme les coefficients  $V_j$  et  $W_j$  de manière à obtenir des signaux qui correspondent au smooth ( $S_j$ ) et au détail ( $D_j$ ) à cette échelle.

En fait, on peut réaliser cette chaîne de traitements en une seule ligne de commande, grâce à la fonction `mra` :

```
> mymra=mra(z, filter="haar", n.levels=4)
> slotNames(mymra)

[1] "D"          "S"          "filter"     "level"     "boundary"  "series"    "class.X"
[8] "attr.X"    "method"
```

```
> # Pour afficher les détails
> mymra@D
> # Pour afficher les smooths
> mymra@S
```

```
> # Pour le détail de niveau 3
> mymra@D$D3
```

```
[,1]
[1,] 1.75
[2,] 1.75
[3,] 1.75
[4,] 1.75
[5,] -1.75
[6,] -1.75
[7,] -1.75
[8,] -1.75
[9,] 2.30
[10,] 2.30
[11,] 2.30
[12,] 2.30
[13,] -2.30
[14,] -2.30
[15,] -2.30
[16,] -2.30
```

```
> # Pour le smooth de niveau 3
> mymra@S$S3
```

```
[,1]
[1,] 0.300
[2,] 0.300
[3,] 0.300
[4,] 0.300
[5,] 0.300
[6,] 0.300
[7,] 0.300
[8,] 0.300
[9,] 1.475
[10,] 1.475
[11,] 1.475
[12,] 1.475
[13,] 1.475
[14,] 1.475
[15,] 1.475
[16,] 1.475
```

On vérifie que les smooths correspondent aux moyennes locales aux différentes échelles et que les détails correspondent aux variations locales aux différentes échelles.

### 1.3 Quelques éléments de vocabulaire

**Ondelette** : il s'agit de la fonction utilisée pour la décomposition en ondelettes. Il en existe plusieurs sortes, par exemple l'ondelette de Haar, l'ondelette en chapeau mexicain, l'ondelette LA(8) (pour "least asymmetric of length 8), etc.)

**Coefficients d'échelle** (scaling coefficients) de niveau  $j$  :  $V_j$  : ce sont des séries de valeurs qui reflètent les moyennes locales de  $Z$  à l'échelle  $2^j \Delta_t$ . On peut définir  $V_1, V_2, \dots, V_J$ . Le nombre d'éléments de  $V_j$  dépend de  $j$  ( $N/2^j$  éléments).

**Coefficients d'ondelettes** (wavelet coefficients) de niveau  $j$  :  $W_j$  : ce sont des séries de valeurs qui reflètent les variations locales de  $Z$  à l'échelle  $2^j \Delta_t$ . On peut définir  $W_1, W_2, \dots, W_J$ . Le nombre d'éléments de  $W_j$  dépend de  $j$  ( $N/2^j$  éléments).

**Smooth** de niveau  $j$  : ce sont des séries de valeurs assimilables aux moyennes locales de  $Z$  à l'échelle  $2^j \Delta_t$ . On peut définir  $S_1, S_2, \dots, S_J$ .  $S_j$  est de même longueur que  $Z$  quel que soit  $j$ .

**Détail** de niveau  $j$  : ce sont des séries de valeurs assimilables aux variations locales de  $Z$  à l'échelle  $2^j \Delta_t$ . On peut définir  $D_1, D_2, \dots, D_J$ .  $D_j$  est de même longueur que  $Z$  quel que soit  $j$ .

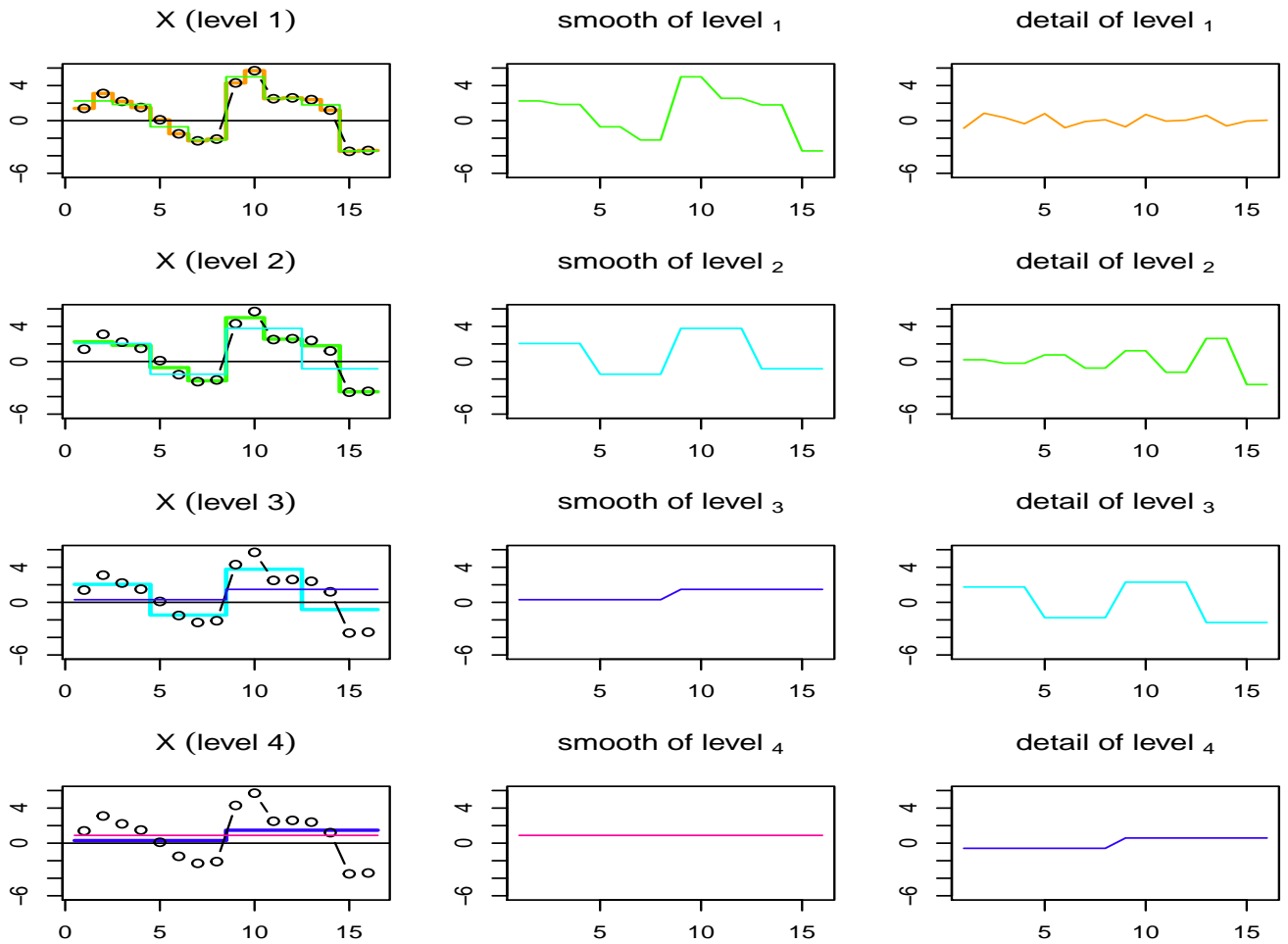


FIGURE 7 – Analyse multi-résolution d'un signal  $Z$  : les différentes lignes correspondent aux différents niveaux de décomposition. A gauche, on a représenté le signal brut (ronds noirs) ainsi que les moyennes locales propres à une échelle donnée.

# Chapitre 2

## Approfondissements techniques

Cette partie est, comme son nom l'indique, assez technique. Elle n'est pas indispensable pour comprendre le principe et les utilisations possibles de la décomposition en ondelettes discrètes, néanmoins elle pourra s'avérer utile, par la suite, lorsque l'utilisation de la décomposition en ondelettes suscitera chez vous quelques interrogations.

### 2.1 Différents types d'ondelettes

Dans la première section, on a abordé les ondelettes via l'exemple des ondelettes de Haar, en montrant qu'appliquer la transformation en ondelettes consistait en quelque sorte à calculer des moyennes et des variations locales à différentes échelles. On parle en fait des ondelettes ou fonctions d'ondelettes, comme de "filtres".

Revenons sur cette notion ici.

De manière générale, on peut décrire un filtre comme étant un dispositif à travers lequel on fait passer quelque chose pour n'en retenir qu'une partie. Ici, le "quelque chose" est un signal, le "dispositif" est une fonction, et la "partie" est un ensemble de coefficients (i.e. un nouveau signal).

Ainsi, appliquer le filtre d'échelle et le filtre d'ondelettes de Haar à un signal  $Z$ , au niveau 1, revient à utiliser sur lui, en un emplacement  $i$ , les fonctions  $g$  et  $h$ . Ces fonctions sont définies respectivement par les coefficients ( $g_1 \approx 0.71, g_2 \approx 0.71$ ) et ( $h_1 \approx 0.71, h_2 \approx -0.71$ ).

La manière dont on utilise ces coefficients sur le signal correspond à ce qu'on appelle une "convolution". En gros, cela revient à considérer l'image en miroir de  $g$  et  $h$  (on parle de "filtre miroir en quadrature"), puis

- à multiplier un à un les éléments du signal et du filtre miroir en quadrature
- à sommer l'ensemble pour obtenir un coefficient d'échelle (pour le filtre  $g$ ) ou un coefficient d'ondelettes (pour le filtre  $h$ ) à un emplacement  $i$
- à décaler le filtre miroir en quadrature pour recommencer à l'emplacement suivant

La notation mathématique (pour notre signal  $z$  de longueur 16, dans le cas de l'utilisation de l'ondelette de Haar) donne ceci :

$$\forall i \in \llbracket 1, 8 \rrbracket \quad V_{1,i} = g_2 * x_i + g_1 * x_{i+1} \quad (2.1)$$

$$\forall i \in \llbracket 1, 8 \rrbracket \quad W_{1,i} = h_2 * x_i + h_1 * x_{i+1} \quad (2.2)$$

Ces coefficients définissant les filtres  $g$  et  $h$  sont représentés dans la figure 8. On peut avoir accès à leurs valeurs via les commandes suivantes :

```
> filter=wt.filter("haar")
> filter@g
```

```
[1] 0.7071068 0.7071068
```

```
> filter@h
```

```
[1] 0.7071068 -0.7071068
```

Rappelons que pour ce signal  $z$ ,  
on a :

```
z1 = 1.4
z2 = 3.1
z3 = 2.2
z4 = 1.5
z5 = 0.1
z6 = -1.5
z7 = -2.3
z8 = -2.1
z9 = 4.3
z10 = 5.7
z11 = 2.5
z12 = 2.6
z13 = 2.4
z14 = 1.2
z15 = -3.5
z16 = -3.4
```

Autrement dit, on a les coefficients d'échelle suivants au niveau 1 :

```
V1,1 = 0.71 * 1.4 + 0.71 * 3.1
      = 3.18
V1,2 = 0.71 * 2.2 + 0.71 * 1.5
      = 2.62
V1,3 = 0.71 * 1.4 + 0.71 * 3.1
      = -0.99
V1,4 = 0.71 * -2.3 + 0.71 * -2.1
      = -3.11
V1,5 = 0.71 * 4.3 + 0.71 * 5.7
      = 7.07
V1,6 = 0.71 * 2.5 + 0.71 * 2.6
      = 3.61
V1,7 = 0.71 * 2.4 + 0.71 * 1.2
      = 2.55
V1,8 = 0.71 * -3.5 + 0.71 * -3.4
      = -4.88
```

Ce que vous pouvez vérifier  
comme suit :

```
> mydwt=dwt(z,filter="haar")
> mydwt@V$V1
      [,1]
[1,] 3.1819805
[2,] 2.6162951
[3,] -0.9899495
[4,] -3.1112698
[5,] 7.0710678
[6,] 3.6062446
[7,] 2.5455844
[8,] -4.8790368
```

A nouveau, on a pour ce signal  
 $z$  :

```
z1 = 1.4
z2 = 3.1
z3 = 2.2
z4 = 1.5
z5 = 0.1
z6 = -1.5
z7 = -2.3
z8 = -2.1
z9 = 4.3
z10 = 5.7
z11 = 2.5
z12 = 2.6
z13 = 2.4
z14 = 1.2
z15 = -3.5
z16 = -3.4
```

Et l'on a les coefficients d'ondelettes suivants au niveau 1 :

```
W1,1 = -0.71 * 1.4 + 0.71 * 3.1
      = 1.20
W1,2 = -0.71 * 2.2 + 0.71 * 1.5
      = -0.49
W1,3 = -0.71 * 1.4 + 0.71 * 3.1
      = -1.13
W1,4 = -0.71 * -2.3 + 0.71 * -2.1
      = 0.14
W1,5 = -0.71 * 4.3 + 0.71 * 5.7
      = 0.99
W1,6 = -0.71 * 2.5 + 0.71 * 2.6
      = 0.07
W1,7 = -0.71 * 2.4 + 0.71 * 1.2
      = -0.85
W1,8 = -0.71 * -3.5 + 0.71 * -3.4
      = 0.07
```

Ce que vous pouvez vérifier  
comme suit :

```
> mydwt=dwt(z,filter="haar")
> mydwt@W$W1
      [,1]
[1,] 1.20208153
[2,] -0.49497475
[3,] -1.13137085
[4,] 0.14142136
[5,] 0.98994949
[6,] 0.07071068
[7,] -0.84852814
[8,] 0.07071068
```

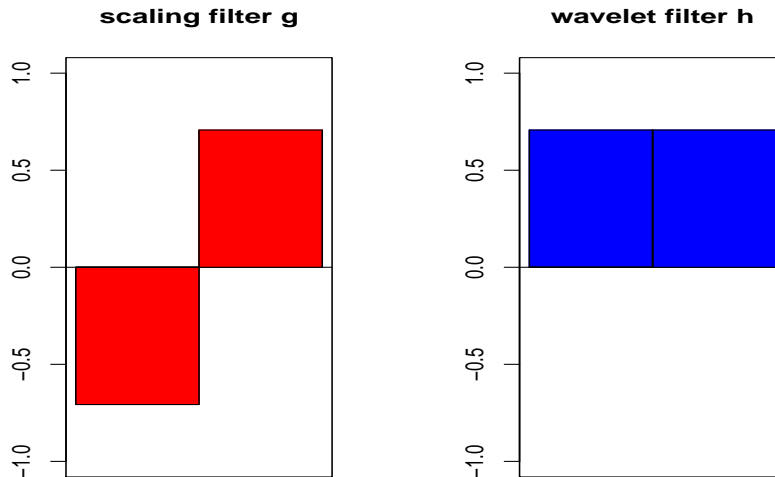


FIGURE 8 – Filtres d'échelle et d'ondelettes pour l'ondelette de Haar (ici filtres miroirs en quadrature).

Vous pouvez constater que l'utilisation de ces filtres miroirs en quadrature agissent bien de telle manière qu'ils renvoient des coefficients d'échelle et des coefficients d'ondelettes proportionnels à une moyenne locale d'une part, et à une variation locale d'autre part.

Ainsi, de manière générale, tout filtre d'échelle ou d'ondelettes, quel que soit son type, se définit par un ensemble de coefficients  $(g_1, g_2, \dots, g_L)$   $(h_1, h_2, \dots, h_L)$ .

Le package wavelets vous propose un certain nombre d'ondelettes, qui se distinguent les unes des autres par leur type et leur longueur. Les types disponibles sont "Daubechies" (l'ondelette de Haar est en fait une ondelette de type Daubechies, de longueur 2), "Least Asymmetric", "Best Localized", et "Coiflet".

Elles sont nommées de la manière suivante dans le package "wavelets" :

- Daubechies : "haar", "d4", "d6", "d8", "d10", "d12", "d14", "d16", "d18", "d20"
- Least Asymmetric : "la8", "la10", "la12", "la16", "la18", "la20"
- Best Linearized : "bl14", "bl18", "bl20"
- Coiflet : "c6", "c12", "c18", "c24", "c30"

Choisir son filtre d'ondelette n'est pas anodin (même s'il résulte souvent, en fin de compte, d'une décision assez empirique), puisque les résultats de la décomposition en dépendent. Voir (Percival and Walden, 2006) (pp135-136) pour plus de détails sur le sujet.

Notez que pour les filtres de longueur supérieure à 2, les "fenêtres" à travers lesquelles on analyse le signal se chevauchent partiellement.

## 2.2 Taille du signal et niveau de décomposition maximal

Le niveau maximal de décomposition  $J_{max}$  dépend de la taille du signal et du filtre d'ondelette utilisé.

$$J_{max} \leq \frac{\ln\left(\frac{n-1}{L-1} + 1\right)}{\ln(2)}$$

Par défaut, la fonction "dwt" effectuera la décomposition au niveau maximal. On peut le vérifier comme suit :



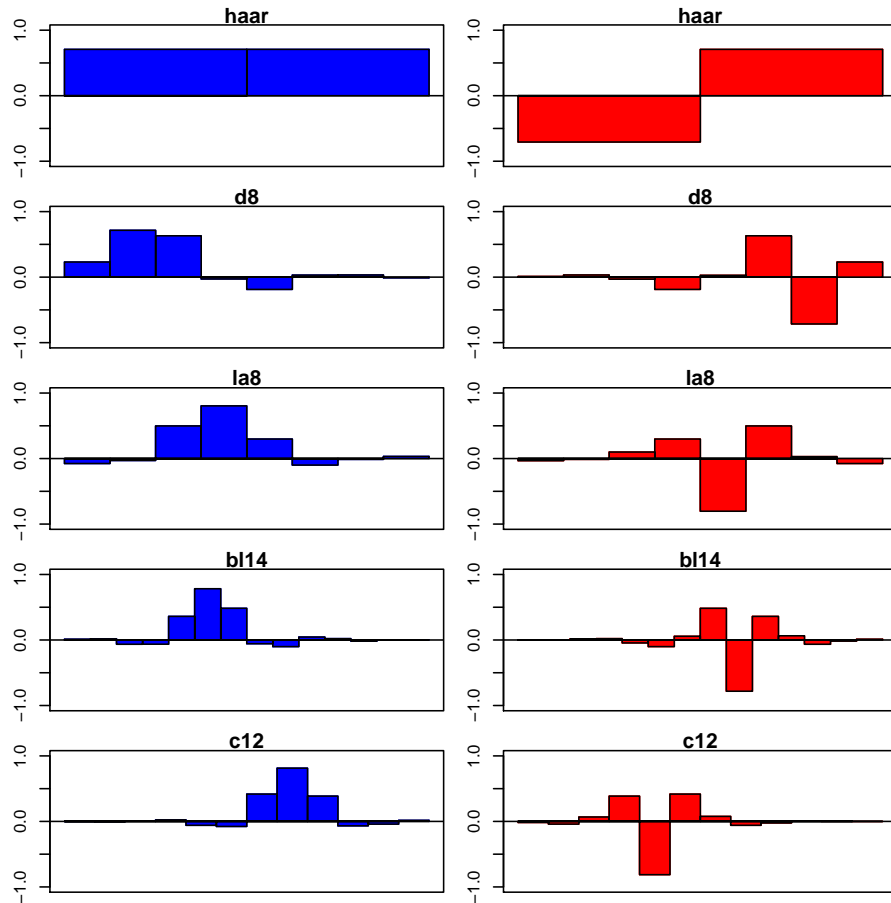


FIGURE 9 – Quelques autres exemples de filtres d'échelle et ondelettes (ici filtres miroirs en quadrature).

```
> n=4234
> L=8
> Jmax=function(n,L){return(floor(log(((n - 1)/(L - 1)) + 1)/log(2)))}
> Jmax(n,L)
```

[1] 9

```
> mydwt=dwt(rnorm(n,0,1),filter="la8")
> mydwt@level
```

[1] 9

>

Rien n'empêche cependant d'effectuer la décomposition que jusqu'à un niveau inférieur :

```
> mydwt=dwt(rnorm(n,0,1),filter="la8",n.levels=3)
> mydwt@level
```

[1] 3

## 2.3 Taille des signaux et problèmes sur les bords

Depuis le début nous avons choisi de prendre l'exemple d'un signal de longueur  $n = 16$ . Découper le signal en segments de longueur 2, 4, 8 ou 16 ne posait donc aucun problème. Que se passe-t-il pour un signal d'une longueur qui n'est pas ainsi découposable ?

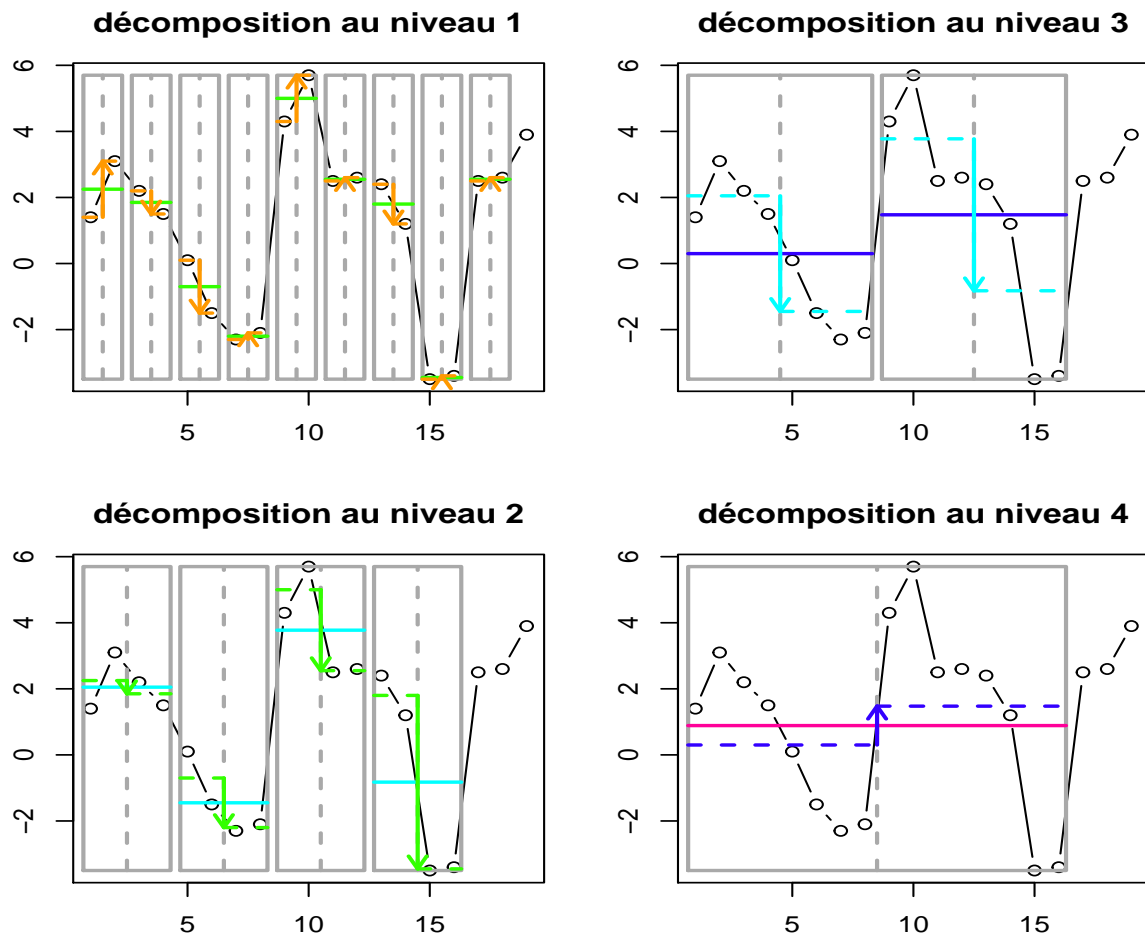


FIGURE 10 – Un exemple de problème en bordure lié à la taille du signal : la description se retrouve "tronquée" à droite

On utilise une méthode qui consiste à "étendre artificiellement" le signal pour travailler sur un signal d'une longueur propre à éviter tout problème. Pour ce faire, on procède soit de manière périodique, soit de manière réflexive (cf figure 11, qui illustre les deux méthodes sur un signal de longueur  $n = 19$ ).

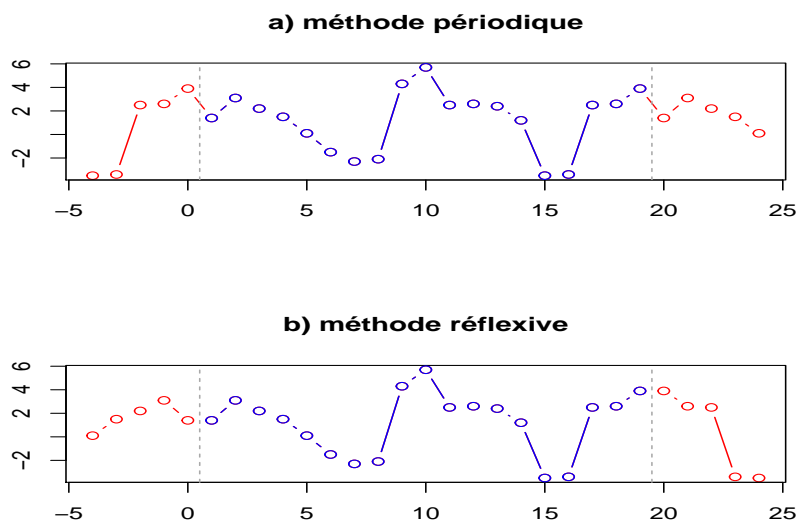


FIGURE 11 – Illustration des deux méthodes de bordure. Le signal en lui-même est en bleu, les bordures sont en rouge. a) Méthode périodique : tout se passe comme si le signal s'intégrait dans une répétition périodique du même signal, i.e. la bordure gauche correspond aux derniers points du signal, la bordure droite correspond aux premiers points du signal. b) Méthode réflexive : tout se passe comme si les droites pointillées verticales étaient deux miroirs : la bordure gauche correspond au reflet des premiers points du signal, la bordure droite correspond au reflet des derniers points du signal.

## Chapitre 3

# Première utilisation possible : la décomposition de la variance ou de l'énergie

### 3.1 Principe

On considère la définition de la norme d'un vecteur  $Z$  de longueur  $n$  :

$$\|Z\| = \sqrt{\sum_{i=1}^n Z_i^2} \quad (3.1)$$

La norme quantifie la somme des écarts absolus des éléments de  $Z$  à 0 ; c'est donc une métrique qui quantifie **la variabilité de  $Z$  autour de 0**.

**Remarque :** la variance se distingue de la norme en quantifiant la variabilité de  $Z$  autour de sa moyenne  $\bar{Z}$  et non autour de 0 :

$$Var(Z) = \frac{1}{n-1} \sum_{i=1}^n (Z_i - \bar{Z})^2 \quad (3.2)$$

Les coefficients d'échelle et d'ondelettes ont la propriété suivante :

$$\|X\|^2 = \|W_1\|^2 + \|V_1\|^2 \quad (3.3)$$

$$= \|W_1\|^2 + \|W_2\|^2 + \|V_2\|^2 \quad (3.4)$$

$$= \|W_1\|^2 + \|W_2\|^2 + \|W_3\|^2 + \|V_3\|^2 \quad (3.5)$$

$$= \|W_1\|^2 + \|W_2\|^2 + \dots + \|W_{Jmax}\|^2 + \|V_{Jmax}\|^2 \quad (3.6)$$

Ce que l'on peut aussi écrire de la manière suivante

$$\forall j \in \llbracket 1 : Jmax \rrbracket \quad \|X\|^2 = \sum_{j=1}^J \|W_j\|^2 + \|V_j\|^2 \quad (3.7)$$

Ce que signifient les équations ci-dessus, c'est que **la variabilité de  $X$  autour de 0 peut être décomposée en une somme de variabilités aux différentes échelles**.

## 3.2 Exemple

La figure 12 vous montre quelques exemples de signaux un peu caricaturaux pour vous permettre de mieux saisir l'information apportée par la décomposition de l'énergie :

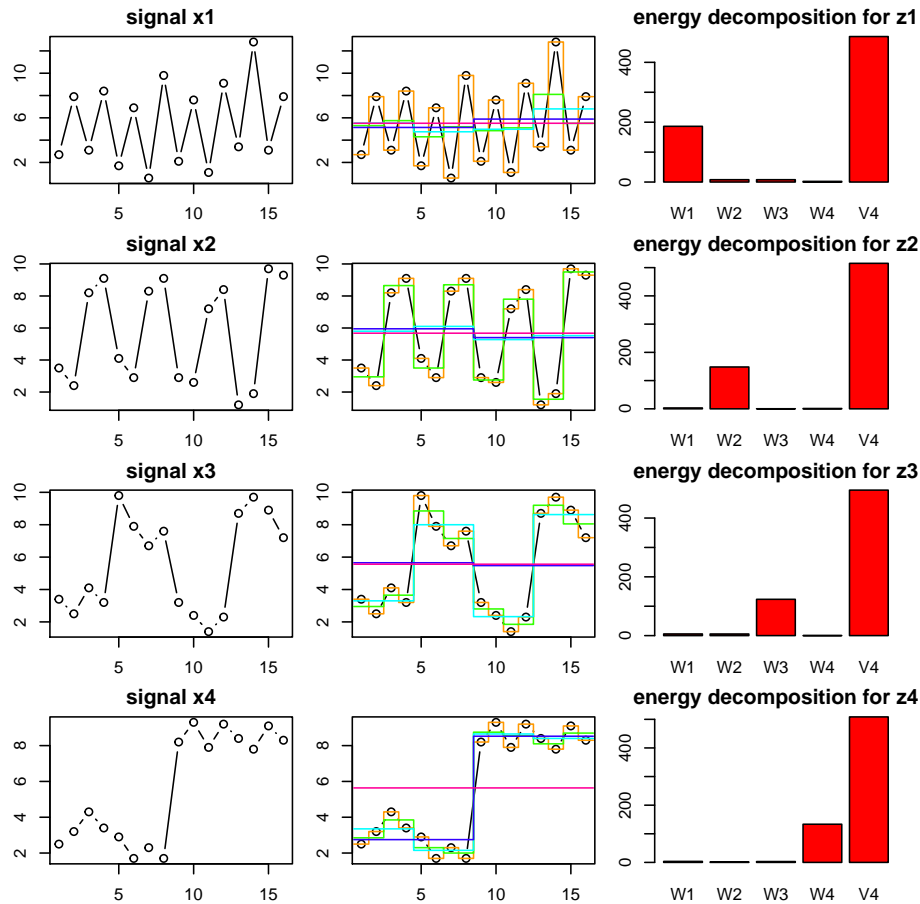


FIGURE 12 – Des exemples de décomposition de la variance

- Dans le premier cas, **le signal varie beaucoup d'un point à l'autre**, mais les moyennes locales sur 2 points (ou davantage) varient peu. Cela se traduit par une énergie plus grande au niveau 1 (norme coefficients d'échelle de niveau 1,  $W_1$ ).
- Dans le deuxième cas, le signal varie peu entre deux points adjacents. De même, les moyennes locales sur 4 et 8 points sont quant à elles relativement stables. Par contre **les moyennes locales sur 2 points varient beaucoup entre elles**. Cela se traduit par une énergie plus grande au niveau 2 (norme des coefficients d'échelle de niveau 2,  $W_2$ ).
- Dans le troisième cas, **ce sont surtout les moyennes locales sur 4 points qui varient beaucoup**, alors que les moyennes locales aux autres échelles sont relativement stables. Cela se traduit par une énergie plus grande au niveau 3 (norme coefficients d'échelle de niveau 3,  $W_3$ ).
- Dans le quatrième cas, **ce sont surtout les moyennes locales sur 8 points qui varient beaucoup**, alors que les moyennes locales aux autres échelles sont relativement stables. Cela se traduit par une énergie plus grande au niveau 4 (norme des coefficients d'échelle de niveau 3,  $W_4$ ).

Revenons maintenant à notre premier exemple de signal Z, pour lequel les interprétations sont moins tranchées :

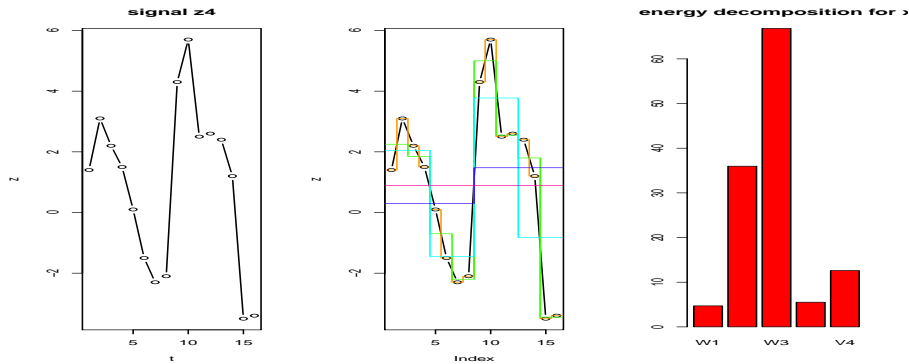


FIGURE 13 – Décomposition de la variance pour le signal Z

Ici, les niveaux auxquels la variabilité est la plus forte sont les niveaux 2 et 3. On pourrait se servir de ce constat, par exemple, pour débruiter le signal, en considérant que ce qui se passe au niveau d'échelle 1 et 4 est peu important (cf section 5)

### 3.3 Mise en oeuvre sous R

```
> z=c(1.4,3.1,2.2,1.5,0.1,-1.5,-2.3,-2.1,
+     4.3,5.7,2.5,2.6,2.4,1.2,-3.5,-3.4)
> mydwt=dwt(z,"haar")
> E=f(z)
> W_E1=f(mydwt@W$W1)
> W_E2=f(mydwt@W$W2)
> W_E3=f(mydwt@W$W3)
> W_E4=f(mydwt@W$W4)
> V_E1=f(mydwt@V$V1)
> V_E2=f(mydwt@V$V2)
> V_E3=f(mydwt@V$V3)
> V_E4=f(mydwt@V$V4)
> # On vérifie la propriété de décomposition de l'énergie:
> print(E)

[1] 125.62

> print(W_E1+V_E1)

[1] 125.62

> print(W_E1+W_E2+V_E2)

[1] 125.62

> print(W_E1+W_E2+W_E3+V_E3)

[1] 125.62

> print(W_E1+W_E2+W_E3+W_E4+V_E4)
```

```

[1] 125.62

> # On peut afficher des résultats de décomposition de l'énergie
> # avec leurs valeurs brutes ou bien en pourcentages:
> print(c(W_E1,W_E2,W_E3,W_E4,V_E4))

[1] 4.7000 35.9750 66.8200 5.5225 12.6025

> print(c(W_E1,W_E2,W_E3,W_E4,V_E4)/E*100)

[1] 3.741442 28.637956 53.192167 4.396195 10.032240

> # pour obtenir le même type de graphique que ci-dessus
> # mais en pourcentages:
> barplot(c(W_E1,W_E2,W_E3,W_E4,V_E4)/E*100,
+         names.arg = c("W1","W2","W3","W4","V4"),
+         col="red",
+         main="energy decomposition for z")

```

Pour aller plus vite, je vous propose en fait d'utiliser la fonction "var\_decompo", que vous trouverez dans le fichier "wavelets\_tools.R".

## Chapitre 4

# Deuxième utilisation possible : l'analyse multi-résolution(MRA : multi-resolution analysis)

### 4.1 Principe

La transformation en ondelettes discrètes est conçue pour permettre la décomposition d'un signal  $Z$  en  $J+1$  signaux, apportant une information propre à leur échelle :

$$Z = \sum_{j=1}^J D_j + S_J \quad (4.1)$$

où  $D_j$  est le détail de niveau  $j$  et  $S_J$  est dit "smooth" d'ordre  $J$ .

### 4.2 Exemple

En anglais, on parle de "MultiResolution Analysis" (MRA), d'où le nom de la fonction qui calcule cette transformation sous R :

```
> z=c(1.4,3.1,2.2,1.5,0.1,-1.5,-2.3,-2.1,
+      4.3,5.7,2.5,2.6,2.4,1.2,-3.5,-3.4)
> mymra=mra(z, filter="haar", n.levels=4)
> slotNames(mymra)
```

```
[1] "D"      "S"      "filter" "level"  "boundary" "series"  "class.X"
[8] "attr.X" "method"
```

```
> mymra@D$D1
      [9,] -0.70
     [10,]  0.70
      [,1]
[1,] -0.85
[2,]  0.85
[3,]  0.35
[4,] -0.35
[5,]  0.80
[6,] -0.80
[7,] -0.10
[8,]  0.10
      [9,] -0.70
     [10,]  0.70
     [11,] -0.05
     [12,]  0.05
     [13,]  0.60
     [14,] -0.60
     [15,] -0.05
     [16,]  0.05
> mymra@D$D2
```



	[,1]		[9,]	1.225
[1,]	0.200		[10,]	1.225
[2,]	0.200		[11,]	-1.225
[3,]	-0.200		[12,]	-1.225
[4,]	-0.200		[13,]	2.625
[5,]	0.750		[14,]	2.625
[6,]	0.750		[15,]	-2.625
[7,]	-0.750		[16,]	-2.625
[8,]	-0.750			

Remarquez les égalités entre les signaux suivants :

$$Z = D_1 + S_1 \tag{4.2}$$

$$= D_1 + D_2 + S_2 \tag{4.3}$$

$$= D_1 + D_2 + D_3 + S_3 \tag{4.4}$$

$$= D_1 + D_2 + D_3 + D_4 + S_4 \tag{4.5}$$

```
> signal_1=mymra@D$D1+mymra@S$S1
> signal_2=mymra@D$D1+mymra@D$D2+mymra@S$S2
> signal_3=mymra@D$D1+mymra@D$D2+mymra@D$D3+mymra@S$S3
> signal_4=mymra@D$D1+mymra@D$D2+mymra@D$D3+mymra@D$D4+mymra@S$S4
> data.frame(z,s1=signal_1,s2=signal_2,s3=signal_3,s4=signal_4)
```

	z	s1	s2	s3	s4
1	1.4	1.4	1.4	1.4	1.4
2	3.1	3.1	3.1	3.1	3.1
3	2.2	2.2	2.2	2.2	2.2
4	1.5	1.5	1.5	1.5	1.5
5	0.1	0.1	0.1	0.1	0.1
6	-1.5	-1.5	-1.5	-1.5	-1.5
7	-2.3	-2.3	-2.3	-2.3	-2.3
8	-2.1	-2.1	-2.1	-2.1	-2.1
9	4.3	4.3	4.3	4.3	4.3
10	5.7	5.7	5.7	5.7	5.7
11	2.5	2.5	2.5	2.5	2.5
12	2.6	2.6	2.6	2.6	2.6
13	2.4	2.4	2.4	2.4	2.4
14	1.2	1.2	1.2	1.2	1.2
15	-3.5	-3.5	-3.5	-3.5	-3.5
16	-3.4	-3.4	-3.4	-3.4	-3.4

Pour faciliter l'utilisation des résultats de l'analyse multi-résolution, vous pouvez utiliser la fonction "sum\_signals" du fichier "wavelets\_tools.R" de la manière suivante :

```
> signal_4=sum_signals(mymra,"D1+D2+D3+D4+S4")
> # équivalent à:
> signal_4=sum_signals(mymra, levelmin=1, levelmax=4)
> # équivalent à :
> signal_4=mymra@D$D1+mymra@D$D2+mymra@D$D3+mymra@D$D4+mymra@S$S4
> signal_3a4=sum_signals(mymra,"D3+D4+S4")
> # équivalent à:
> signal_3a4=sum_signals(mymra, levelmin=3, levelmax=4)
> # équivalent à :
> signal_3a4=mymra@D$D3+mymra@D$D4+mymra@S$S4
```

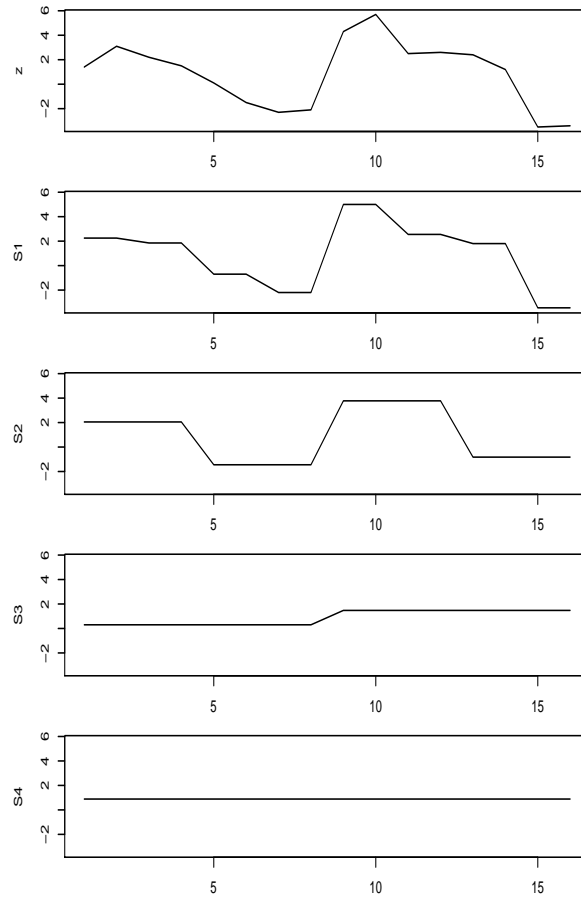


FIGURE 14 – Multiresolution analysis

### 4.3 Mise en oeuvre sous R

La figure 14 peut être générée à l'aide de la fonction `plot_mra` que vous trouverez dans le fichier "wavelets\_tools.R" :

```
> plot_mra(mymra,levels=c("S1","S2","S3","S4"))
```

Vous pouvez également imaginer d'autres combinaisons des sous-signaux, par exemple :

```
> # Pour représenter le signal, et des approximations de plus en plus grossières du signal:
> plot_mra(mymra,levels=c("z","S4+D4+D3+D2+D1","S4+D4+D3+D2","S4+D4+D3","S4+D4","S4"))
> # Pour reproduire la représentation "usuelle" de la décomposition en ondelettes:
> # Z=S4+D4+D3+D2+D1
> plot_mra(mymra,levels=c("S4","D4","D3","D2","D1","z"))
> # Pour agréger les détails aux niveaux 1 et 2, et aux niveaux 3 et 4
> # Z=(D1+D2)+(D3+D4)+(S4)
> plot_mra(mymra,levels=c("z","D1+D2","D3+D4","S4"),color="rainbow")
```

## Chapitre 5

# Troisième utilisation possible : Le débruitage et/ou la compression des signaux

En décomposant les signaux à diverses échelles, les ondelettes permettent d'isoler une partie du signal jugée non-informative (par exemple ses variations à très petite échelle, considérées comme du bruit).

Ainsi, elles peuvent permettre de **retirer du signal cette partie non-informative, en le reconstruisant à partir des détails à certaines échelles choisies**, autrement dit de **débruiter le signal d'origine**.

Considérons ainsi le signal représenté dans la figure 15.

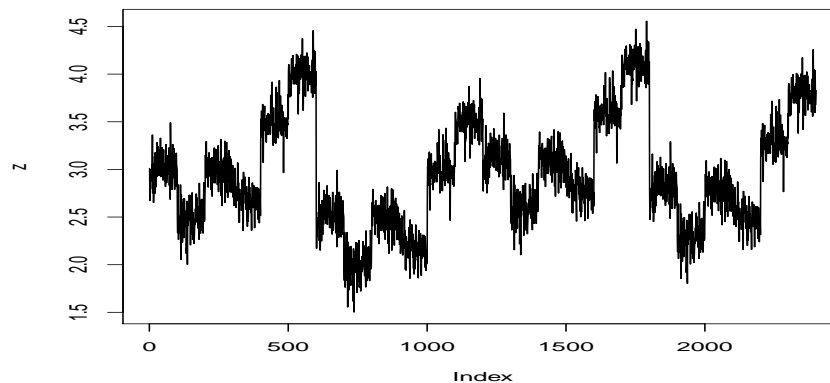


FIGURE 15 – Un exemple de signal bruité à petite échelle Z

Il semble que la variabilité aux plus petites échelles soit assez négligeable par rapport à des tendances à plus large échelle. Pour vérifier cela on peut utiliser le principe de décomposition de la variance qu'on a abordé précédemment, cf figure ??, que l'on crée comme suit :

```
> mydwt=dwt(z,"haar")
> var_decompo(mydwt)
```

On peut ainsi choisir de ne s'intéresser qu'aux niveaux d'échelles 6 à 10, en considérant tout ce qui se passe aux échelles plus fines comme du bruit.

Dès lors, il est possible de **débruiter le signal en le reconstruisant à partir des détails de niveaux 6 à 10 et du smooth de niveau 10** comme suit :

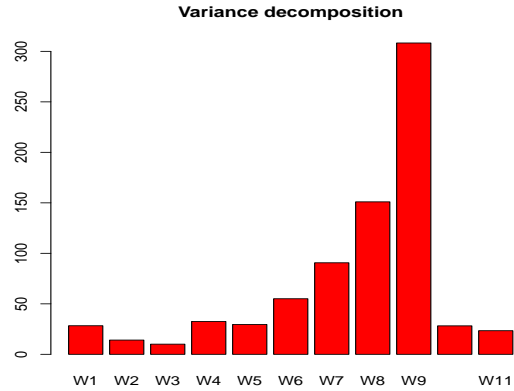


FIGURE 16 – Décomposition de la variance aux différentes échelles du signal Z : la variabilité aux niveaux 1,2,3,4 et 5 est très faible en comparaison de la variabilité aux niveaux 6,7,8, 9 et 10.

FIGURE 17 – vardecompoZbis

```
> mymra=mra(z,n.levels=10,boundary="reflection")
> z_debruite=sum_signals(mymra,levelmin=6,levelmax=6)
> plot(z,type="l")
> points(z_debruite,type="l", col="blue", lwd=2)
```

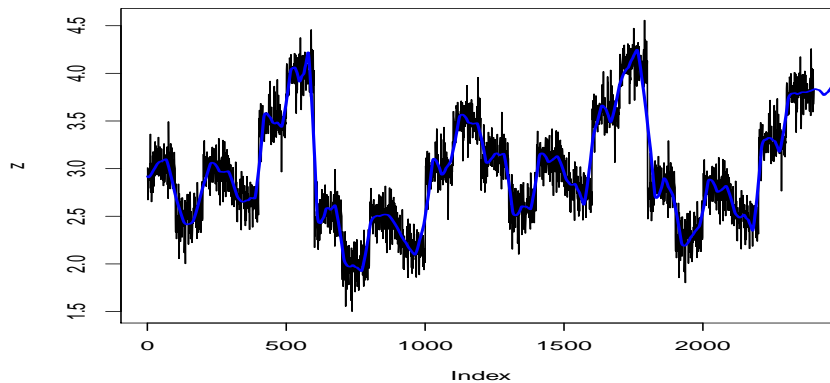


FIGURE 18 – Le signal d'origine (en noir) et le signal débruité (en bleu)

Débruiter le signal revient ici à considérer que les seules informations pertinentes qu'il nous apporte proviennent des niveaux 6 à 10. On est donc en mesure de construire le signal débruité à partir des signaux  $D_6, D_7, D_8, D_9, D_{10}$ , et  $S_{10}$ .

Or, nous avons vu que les détails et smooths de niveau  $j$  pouvaient être déduits mathématiquement des coefficients d'ondelette et d'échelle du même niveau. Au niveau  $j$ , si les détails et les smooths sont aussi longs que le signal d'origine, il y a par contre, au maximum,  $N/2^j$  coefficients d'échelle et  $N/2^j$  coefficients d'ondelette.

- Donc pour un signal de longueur  $n$  (ici  $n=2400$ ), on a
- au maximum  $n/2^6$  (ici 75) coefficients d'ondelette au niveau 6

- au maximum  $n/2^7$  (ici 37) coefficients d'ondelette au niveau 7
- au maximum  $n/2^8$  (ici 18) coefficients d'ondelette au niveau 8
- au maximum  $n/2^9$  (ici 9) coefficients d'ondelette au niveau 9
- au maximum  $n/2^{10}$  (ici 4) coefficients d'ondelette au niveau 10
- au maximum  $n/2^{10}$  (ici 4) coefficients d'échelle au niveau 10

Soit en tout  $75 + 37 + 18 + 9 + 4 + 4 = 147$  coefficients.

On est donc en mesure, avec 147 coefficients, de reconstruire une version débruitée d'un signal qui à l'origine comptait 2400 valeurs. Ces 147 coefficients constituent donc une version compressée du signal d'origine.

## BILAN

Le principe de la décomposition par les ondelettes est le suivant :

Il s'agit en quelque sorte de décomposer un signal  $Z$  en coefficients caractéristiques de son comportement à différentes échelles. On parle de **coefficients d'échelle**  $V_1, V_2, \dots, V_{J_{max}}$  et de **coefficients d'ondelettes**  $W_1, W_2, \dots, W_{J_{max}}$ .

On peut ensuite, à partir de ces coefficients, "reconstruire" des signaux à différents niveaux d'échelle (les **détails**  $D_1, D_2, \dots, D_{J_{max}}$  et les **smooths**  $S_1, S_2, \dots, S_{J_{max}}$ ).  $X$  est en fait la somme des détails et du smooth de plus haut niveau :  $Z = D_1 + D_2 + \dots + D_{J_{max}} + S_{J_{max}}$ .

Les coefficients d'ondelettes et d'échelle permettent de caractériser la variance de  $Z$  aux différents niveaux d'échelle. On parle de **décomposition de la variance**, ou de **décomposition de l'énergie**.

Les détails et les smooths permettent d'analyser  $Z$  à ses différentes échelles. On parle d'**analyse multi-résolution (MRA)**.

On peut enfin débruiter le signal  $Z$  en éliminant ses variations aux échelles les plus fines, c'est-à-dire qu'on s'intéresse à un signal débruité  $Z_{bis}$  qui correspond à la somme des détails et smooth de niveaux les plus élevés.

## Chapitre 6

# Décomposition en ondelettes à chevauchement maximal (Maximum Overlap Discrete Wavelet Transform : MODWT)

### 6.1 Principe

Considérons à nouveau la figure 1. Si l'on s'intéresse par exemple à l'échelle 2 (niveau 1), on peut considérer les segments correspondant aux éléments 1-2, puis 3-4, puis 5-6, etc. mais rien n'indique que considérer ce découpage a plus de sens a priori que considérer des segments qui correspondraient aux éléments 2-3 et 4-5 puis entre 4-5 et 5-6, etc. comme dans la figure 19.

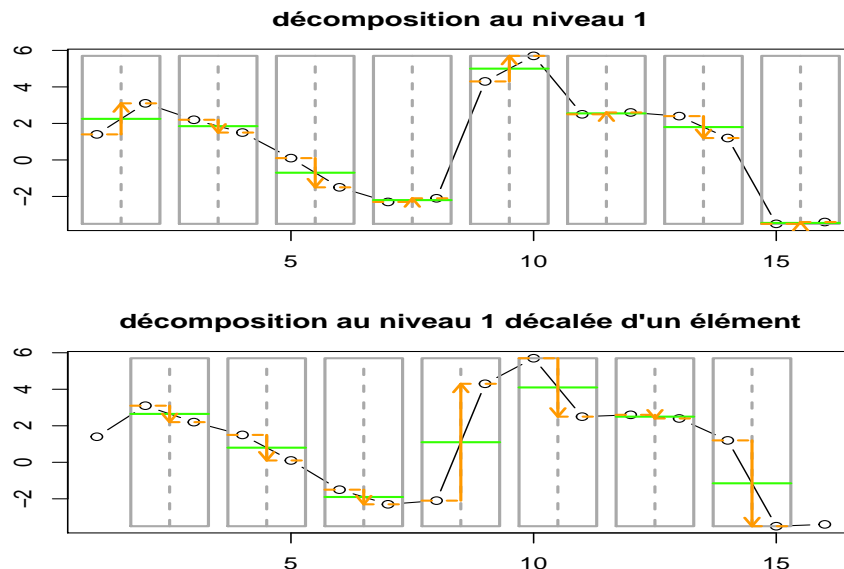


FIGURE 19 – Un exemple de signal simple  $Z$  : on peut pour décrire ce signal au niveau 1 (donc à l'échelle 2), considérer plusieurs manières de découper le signal en secteurs de longueur 2.

De la même manière, si l'on s'intéresse par exemple à l'échelle 4 (niveau 2), on peut considérer les éléments

1-2-3-4, 5-6-7-8, 9-10-11-12, 13-14-15-16. Mais rien ne nous dit là encore que considérer ce découpage a plus de sens que de considérer un même découpage décalé d'un, ou deux, ou trois éléments, comme dans la figure 20.

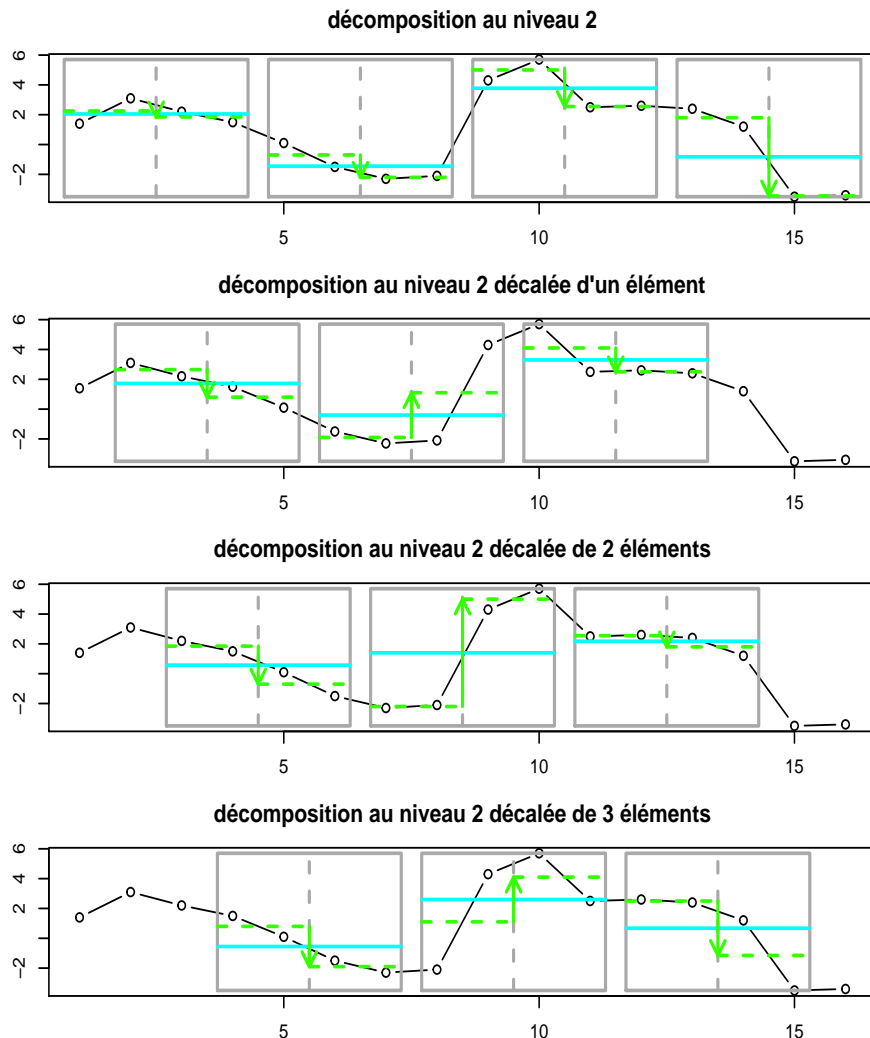


FIGURE 20 – Un exemple de signal simple  $Z$  : on peut pour décrire ce signal au niveau 2 (donc à l'échelle 4), considérer plusieurs manières de découper le signal en secteurs de longueur 4.

Si l'on considère tous ces découpages du signal également pertinents, alors on va chercher à décrire le signal au niveau d'échelle considéré en les prenant tous en compte à la fois. Ainsi, pour décrire ce qui se passe en terme de moyenne locale et de variation locale, on s'intéresse à la **moyenne des moyennes locales** et à la **moyenne des variations locales**

Ainsi, au premier niveau, on considère les moyennes et variations locales suivantes

- sur les segments 1-2, et 2-3 => moyenne et variation locale au point 2
- sur les segments 2-3, et 3-4 => moyenne et variation locale au point 3
- sur les segments 3-4, et 4-5 => moyenne et variation locale au point 4
- sur les segments 4-5, et 5-6 => moyenne et variation locale au point 5
- etc.

Au deuxième niveau, on considère les moyennes et variations locales suivantes

- sur les segments 1 à 4, 2 à 5, 3 à 6, 4 à 7 => moyenne et variation locale au point 4



- sur les segments 2 à 5, 3 à 6, 4 à 7, 5 à 8 => moyenne et variation locale au point 5
- sur les segments 3 à 6, 4 à 7, 5 à 8, 6 à 9 => moyenne et variation locale au point 6
- sur les segments 4 à 7, 5 à 8, 6 à 9, 7 à 10 => moyenne et variation locale au point 7
- etc.

```

> mymra=mra(z,"haar",method="modwt", n.levels=4)
> # moyennes locales au niveau 1:
> mean(c(mean(z[1:2]),mean(z[2:3])))

[1] 2.45

> mean(c(mean(z[2:3]),mean(z[3:4])))

[1] 2.25

> mean(c(mean(z[3:4]),mean(z[4:5])))

[1] 1.325

> mean(c(mean(z[4:5]),mean(z[5:6])))

[1] 0.05

> mymra@S$S1[2:5]

[1] 2.450 2.250 1.325 0.050

> # moyennes locales au niveau 2:
> mean(c(mean(z[1:4]),mean(z[2:5]),mean(z[3:6]),mean(z[4:7])))

[1] 0.95

> mean(c(mean(z[2:5]),mean(z[3:6]),mean(z[4:7]),mean(z[5:8])))

[1] 0.075

> mean(c(mean(z[3:6]),mean(z[4:7]),mean(z[5:8]),mean(z[6:9])))

[1] -0.45625

> mean(c(mean(z[4:7]),mean(z[5:8]),mean(z[6:9]),mean(z[7:10])))

[1] -0.25

> mymra@S$S2[4:7]

[1] 0.95000 0.07500 -0.45625 -0.25000

> # variations locales au niveau 1:
> mean(c(diff(z[1:2]),diff(z[2:3])))

[1] 0.4

> mean(c(diff(z[2:3]),diff(z[3:4])))

[1] -0.8

> mean(c(diff(z[3:4]),diff(z[4:5])))

[1] -1.05

```

```

> mean(c(diff(z[4:5]),diff(z[5:6])))

[1] -1.5

> mymra@D$D1[2:5]

[1] 0.650 -0.050 0.175 0.050

> # moyennes locales au niveau 2:
> mean(c(mean(z[1:4]),mean(z[2:5]),mean(z[3:6]),mean(z[4:7])))

[1] 0.95

> mean(c(mean(z[2:5]),mean(z[3:6]),mean(z[4:7]),mean(z[5:8])))

[1] 0.075

> mean(c(mean(z[3:6]),mean(z[4:7]),mean(z[5:8]),mean(z[6:9])))

[1] -0.45625

> mean(c(mean(z[4:7]),mean(z[5:8]),mean(z[6:9]),mean(z[7:10])))

[1] -0.25

> mymra@S$S2[4:7]

[1] 0.95000 0.07500 -0.45625 -0.25000

>
>

```

Si l'on fait de même pour tous les autres points, alors on obtient des moyennes locales et des variations locales telles que représentées dans la figure 21.

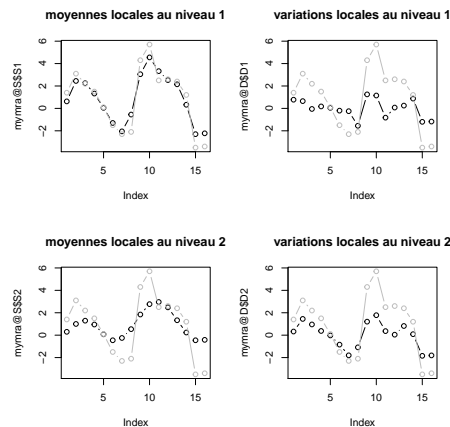


FIGURE 21 – Description de  $z$  aux niveaux d'échelle 1 et 2, avec "chevauchement" des différents découpages. En gris, on a réaffiché le signal  $z$  d'origine.

Remarquez qu'en prenant en compte les chevauchements des différentes fenêtres, alors on devrait obtenir un coefficient pour chaque point de  $Z$  (i.e. les coefficients, à chaque niveau, sont de longueur  $N=16$ )

## 6.2 Mise en oeuvre sous R

La mise en oeuvre de la décomposition en ondelettes à chevauchement maximal se fait à l'aide de la fonction `modwt` :

```
> mymodwt=modwt(z, filter="haar")
```

Remarquez que, de fait, les

# Bibliographie

Percival, D. and A. Walden (2006). *Wavelet Methods for Time Series Analysis*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.