

W.A.W.
(Wavelets At Work)

Christian MADSEN

2 novembre 2006

Table des matières

1	Présentation de l'application	3
1.1	Un exemple pour commencer	3
2	Présentation des Ondelettes	3
2.1	Généralités	3
2.1.1	Conditions pour être une ondelette	4
2.1.2	Transformée en ondelette	4
2.1.3	Reconstruction	4
2.2	Leur utilisation dans l'application	5
2.2.1	Chapeau Mexicain	5
2.2.2	Discrétisation	5
3	Mode d'emploi	5
3.1	Arborescence de l'application	5
3.2	Programme requis	6
3.3	Compilation et exécution de l'application	6
3.4	Paramètres d'exécution du programme	6
3.5	Format des fichiers en entrée	7
3.5.1	Exemple d'un fichier signal	7
3.6	Importance des réglages	7
3.7	Fonctionnalités	7
3.7.1	Chargement d'un fichier de données	7
3.7.2	Enregistrement d'un fichier de données	8
3.7.3	Insertion d'interférences dans le signal	8
3.7.4	Les paramètres	8
4	Analyse de résultats	9
5	Architecture de l'application	11
5.1	Les fichiers de propriétés	11
6	Problèmes rencontrés	12
7	Evolutions envisagées	12
8	Conclusion	12

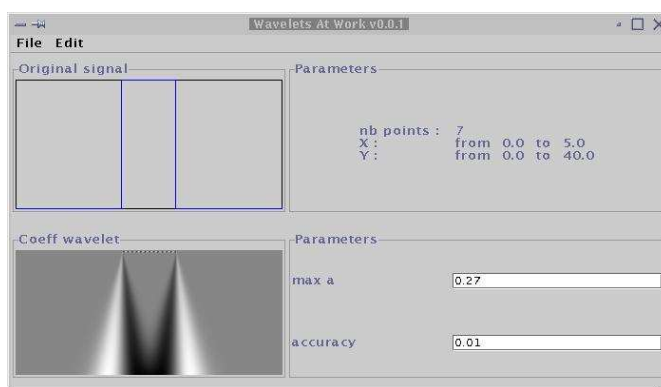
1 Présentation de l'application

WAW est une application qui permet de faire une analyse en ondelettes d'un signal donné.

Ce signal est chargé à partir d'un fichier ascii dont le format est précisé au paragraphe 3.5, l'interface de l'application permet de représenter un graphique de ce signal ainsi que son traitement par une ondelette. Il est possible de modifier un certain nombre de paramètres influant sur la représentation des coefficients en ondelette. Ainsi suivant les réglages on pourra voir apparaître certaines caractéristiques du signal.

1.1 Un exemple pour commencer

Voici un exemple de traitement réalisé avec comme fichier en entrée *waw/data/carre_ok.txt* :



Pour plus de détails sur les options disponibles et les fonctionnalités de l'application, merci de vous reporter au paragraphe 3.

2 Présentation des Ondelettes

2.1 Généralités

Les ondelettes sont des fonctions particulières permettant de dégager des singularités dans un signal.

Leur étude passe tout d'abord par celle de la transformée de Fourier.

Soit $s(t)$ un signal, on considère sa transformée de Fourier tel que

$$\hat{s}(w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-iwt} s(t) dt$$

Malheureusement cette transformée englobe toutes les fréquences du signal ne permettant aucune localisation précise de ses variations. On a donc recours à des fonctions précises (voir conditions au paragraphe 2.1.1) pour calculer des transformées en *temps-fréquence* ou *temps-échelle* d'un signal qui, elles, permettent de localiser les variations de ce signal.

2.1.1 Conditions pour être une ondelette

Certaines conditions sont nécessaires pour obtenir une ondelette, ceci peut paraître *raisonnable* au regard de ses propriétés de détection.

- Tout d'abord la fonction doit être de carré intégrable :

$$\int_{-\infty}^{+\infty} |\psi(t)|^2 dt < \infty$$
- bien localisée,
- et enfin *admissible* ce qui est équivalent à être de *moyenne nulle* car ainsi il y a conservation (à un facteur près) de l'énergie du signal.

2.1.2 Transformée en ondelette

La transformée en ondelette d'un signal $s(t)$ d'énergie finie s'écrit :

$$S(a, b) = \int_{-\infty}^{+\infty} \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) s(t) dt$$

ou encore, en termes des transformées de Fourier :

$$S(a, b) = \sqrt{\frac{a}{2\pi}} \int_{-\infty}^{+\infty} e^{ibw} \overline{\widehat{\psi}(aw)} \widehat{s}(w) dw$$

a représente le coefficient de dilatation (strictement positif), et b le coefficient de translation. C'est en jouant sur ces deux paramètres que l'on obtient une analyse pertinente d'un signal. Le coefficient de translation nous permet de faire *glisser* l'ondelette le long du signal, et le coefficient de dilatation permet de régler la *sensibilité* de l'ondelette aux valeurs du signal.

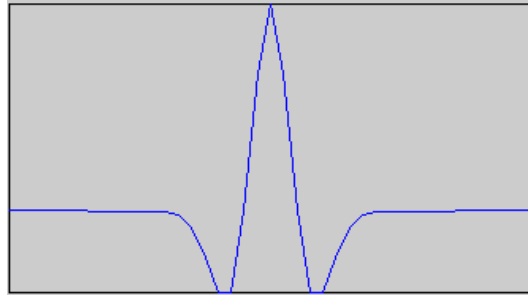
2.1.3 Reconstruction

Un point important de ces coefficients en ondelette est que l'on peut reconstruire le signal original à partir des coefficients pour peu que la discrétisation est suivie certaines règles. Par exemple la discrétisation du demi-plan en *grille dyadique* : $a = 2^{-j}$ et $b = 2^{-j}k$

2.2 Leur utilisation dans l'application

Le programme utilise comme unique ondelette celle dite du *Chapeau Mexicain* (Mexican hat).

2.2.1 Chapeau Mexicain



Cette ondelette est de type réelle.
Elle s'exprime de la façon suivante :

$$\psi_H(x) = (1 - x^2)e^{-x^2/2}$$

et sa transformée de Fourier est

$$\hat{\psi}_H(x) = w^2 e^{-w^2}$$

2.2.2 Discrétisation

La discrétisation de l'intégrale présentée au paragraphe 2.1.2 se fait de façon très simple en remplaçant l'intégrale infinie par une somme finie de termes bornés par les valeurs limites de définition du signal. Par ailleurs cette approximation n'est pas trop gênante car, par définition même, l'ondelette s'écrase rapidement vers 0 (voir exemple au 2.2.1).

3 Mode d'emploi

3.1 Arborescence de l'application

Le répertoire racine de l'application ce nomme **waw/**, il contient un fichier d'aide (*README.TXT*), les fichiers exécutables ainsi que les répertoires suivants :

- **data/** contenant les fichiers de données représentant des signaux et pouvant être chargés à l'aide de l'application.

- `data/images/` contenant des images obtenues à partir des fichiers de signaux.
- `doc/` contenant la documentation java relative à l'application dans le sous-répertoire `javadoc/`; ainsi que le rapport de projet au format `LATEX` dans le sous-répertoire `report/`.
- `javaclass/` contenant les classes java compilées.
- `resources/` contenant les fichiers de ressource utilisés par l'application.
- `src/` contenant les fichiers sources java du programme.

La documentation dite *JavaDoc* relative aux sources du programme peut être générée en se plaçant dans le répertoire `src/` et en tapant la commande :

```
cmadsen@mad:~/prog/waw/src > make doc
```

3.2 Programme requis

WAW est une application écrite en Java, vous devez donc disposer d'une machine virtuelle Java JVM 1.2 (ou supérieure) sur votre machine pour pouvoir l'exécuter.

Vous pourrez obtenir cette machine virtuelle Java pour votre système d'exploitation, gratuitement, sur le site de *Sun* à l'adresse suivante :

<http://java.sun.com/>

3.3 Compilation et exécution de l'application

Pour compiler l'application il faut se placer dans le répertoire `waw/src` puis exécuter la commande `make`.

```
cmadsen@mad:~/prog/waw/src > make
```

Cette commande a pour effet de compiler les sources du programme et de placer les fichiers compilés dans le répertoire `javaclass/`.

L'exécutable se trouve dans le répertoire `waw/` et se nomme suivant le système d'exploitation utilisé, `waw` (sous Unix/Linux), ou `waw.bat` (sous Windows).

Donc pour lancer l'application (sous Linux), placez-vous dans le répertoire `waw/` puis tapez `waw`.

```
cmadsen@mad:~/prog/waw > waw
```

3.4 Paramètres d'exécution du programme

L'application supporte les paramètres suivants au démarrage :

`-v` pour afficher le numéro de version de l'application.

-d pour accéder au mode debug.

3.5 Format des fichiers en entrée

Les fichiers de données correspondants aux signaux, sont des fichiers *ascii* qui contiennent par ligne les valeurs d'un point du signal sous la forme : abscisse ordonnée.

3.5.1 Exemple d'un fichier signal

Voici en exemple le fichier de demo fournit dans le répertoire *data/*.

exemple_signal_1.txt

3.6 Importance des réglages

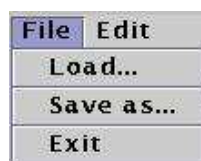
Les paramètres pouvant être modifiés tel que a (coefficient de dilatation), ou le pas de calcul pour l'intégration, sont *déterminants* pour obtenir une image intéressante des coefficients en ondelette. En effet un pas trop grand laisse apparaître des pics correspondant à des variations dans le signal qui n'existe pas. Malheureusement plus le pas est petit plus le temps de calcul est important.

De même pour le paramètre a s'il est trop petit les pics seront trop fins, et s'il est trop grand les pics seront trop gros.

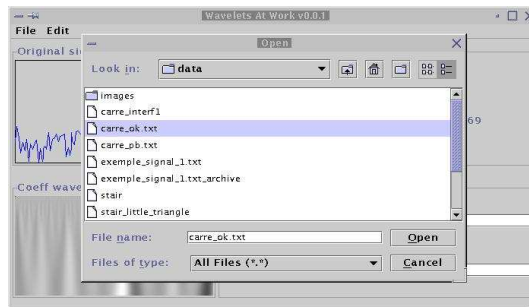
3.7 Fonctionnalités

3.7.1 Chargement d'un fichier de données

Le chargement se fait à l'aide du menu *File* puis *Load...*



Une fenêtre de sélection apparaît alors :



Il ne vous reste plus qu'à choisir parmi les différents signaux.
 Pour créer vos propres signaux, merci de vous reporter au paragraphe 3.5.

3.7.2 Enregistrement d'un fichier de données

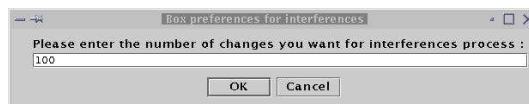
De même que, dans le cas du chargement, une fenêtre permet de choisir l'emplacement du fichier à sauvegarder ainsi que son nom.

3.7.3 Insertion d'interférences dans le signal

L'application permet de générer des variations aléatoires dans un signal.

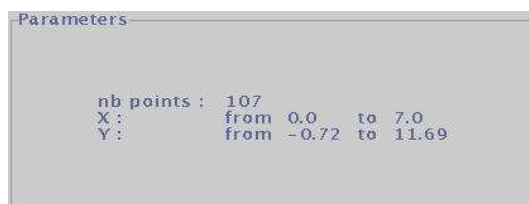


On peut en particulier régler la fréquence de ces interférences (nombre de points à changer dans le signal original) et leur amplitude (valeur maximale du random).

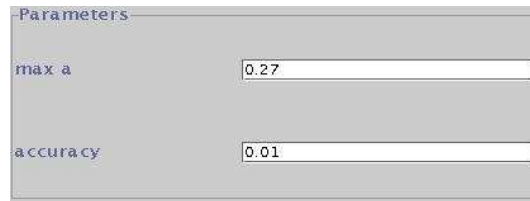


3.7.4 Les paramètres

Du signal : Ils ne peuvent pas être directement modifiés. En revanche ils sont visibles dans la fenêtre principale de l'application :



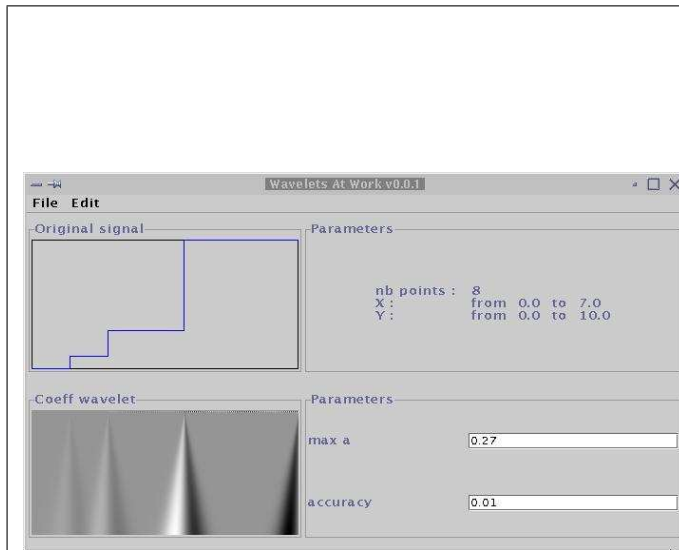
Des coefficients en ondelette : Ils peuvent être directement modifiés. Et le cas échéant cela enclenche le processus de calcul (un compteur en indique la progression en pourcentage).



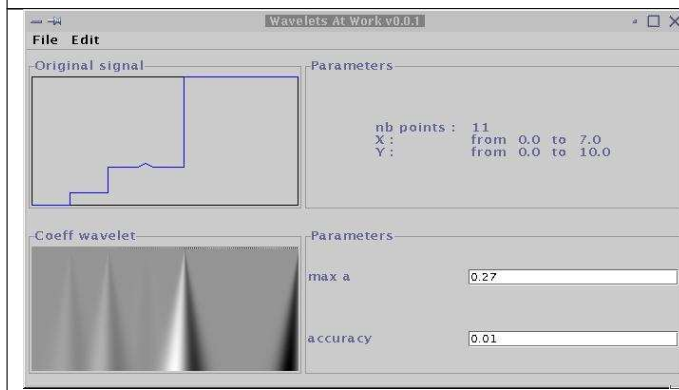
Attention aux réglages (cf 3.6).

4 Analyse de résultats

A screenshot of the "Wavelets At Work v0.0.1" software interface. The window is divided into four panes. The top-left pane shows the "Original signal" as a plot with two vertical lines. The top-right pane shows "Parameters" with "nb points : 7", "X: from 0.0 to 5.0", and "Y: from 0.0 to 40.0". The bottom-left pane shows the "Coeff wavelet" as a plot with two bright, narrow peaks. The bottom-right pane shows "Parameters" with "max a" set to "0.27" and "accuracy" set to "0.01".	<p>Cette première représentation met bien en évidence les deux cônes <i>pointants</i> vers les discontinuités du signal original de façon extrêmement précise. En étudiant de plus près l'image obtenue on constate de légères variations entre ces deux cônes pour les petites valeurs de a (coefficient de dilatation).</p>
--	--



Cette image est la première d'une trilogie qui a pour objectif de montrer la *sensibilité* de l'ondelette par rapport au signal original (ici une fonction en escalier). De façon très nette on remarque un cône plus prononcé au centre de l'image, or il correspond exactement à un brusque saut dans le signal, le plus important du signal. Ceci montre bien que les coefficients en ondelette localisent non seulement les variations mais quantifient aussi leur importance.



Le signal en entrée est le même que précédemment avec l'ajout, sur une de ses phases horizontales, d'un petit signal en triangle. Notez que le cône correspondant à cette perturbation minime n'est quasiment pas visible sur l'image.

	<p>En revanche si l'on augmente la valeur de cette perturbation, alors celle-ci se traduit par un cône bien démarqué (en fait trois : un pour le sommet, et les deux autres pour les bases du triangle). Notez bien que la largeur de la base du triangle, ajouté au signal, est la même que dans l'image précédente.</p>
	<p>Enfin, nous avons ici un exemple du signal en escalier <i>bruité</i> par 100 valeurs avec un écart max au signal de 1,5. On remarque la multitude de pics qui en résulte, avec cependant un cône plus marqué délimitant un brusque saut du signal. On pourrait donc très bien repérer, de façon automatique, cette brusque variation en utilisant une valeur seuil adaptée.</p>

5 Architecture de l'application

Une attention particulière a été portée sur la modularité de l'application ce qui permettra de la faire évoluer simplement.

De plus tout le code java a été documenté en respectant la syntaxe *JavaDoc*.

5.1 Les fichiers de propriétés

Les fichiers de propriétés se trouvent dans le répertoire *resources/* de l'application. L'utilisation des ces fichiers rend l'application **complètement paramétrable**, juste en modifiant un fichier texte, évitant ainsi toute re-

compilation inutile.

De plus ces fichiers permettent à l'application d'être transcrite dans différentes langues de façon très simple.

Voici une description du rôle de chacun d'eux :

- *config.properties* contient tous les paramètres de l'application pouvant être configuré, tel que la taille de la fenêtre les valeurs par défauts des paramètres...
- *info.properties* contient des informations sur l'application qui ne doivent pas être modifiées par l'utilisateur.
- *resources.properties* contient tous les mots et phrases utilisés dans l'application, et définit également l'organisation du menu de l'application.

6 Problèmes rencontrés

- Le calcul de l'intégrale varie suivant le pas d'incrément choisi pour sa résolution. Il en résulte l'apparition de pics alors qu'aucune variation du signal n'est visible et inversement il y a aussi absence de pics là où pourtant le signal varie fortement. Cependant en utilisant un pas faible (< 0.01) on constate que seul les pics les plus cohérents subsistent. Ce problème est également abordé au paragraphe 3.6.

7 Evolutions envisagées

- Pouvoir utiliser différents types d'ondelettes, et plus particulièrement celles décrites par des nombres complexes (par exemple l'ondelette de Morlet). On pourrait alors visualiser le module et la phase des coefficients en ondelettes ainsi obtenus.
- Reconstruire le signal à partir des coefficients en ondelettes. Avec possibilité d'appliquer des valeurs seuil et ainsi d'influer sur le signal reconstruit, avec pour objectif soit la détection des singularités soit leur élimination.
- Prendre comme fichier en entrée des fichiers son.

8 Conclusion

Les ondelettes ouvrent les portes de domaines extrêmement variés comme la détection, le filtrage, la compression, sur des signaux, des images, des

sons...

Leur champs d'action tend à les rendre de plus en plus *populaires* et par conséquent les applications les utilisant sont de plus en plus nombreuses, parmi celles-ci, on peut noter les logiciels de reconnaissance vocale où les ondelettes interviennent dans le découpage des mots, les logiciels médicaux dans lesquels on se sert de leur aptitude à la détection ce qui permet de rendre certains phénomènes *visibles*. On pourrait aussi citer tous les logiciels de compression video et sonore en plein essor actuellement.

Les ondelettes constituent donc un outil mathématique majeur dans les logiciels du présent et du futur.