

# ウェーブレット(Wavelet)変換～基礎編

## はじめに

このワークシートでは、ウェーブレット(Wavelet)の基礎に関して、Mapleによる操作方法、簡単な応用例が紹介されています。

ウェーブレット(Wavelet)変換は、小さい波を拡大縮小と平行移動を適用することで、与えられた入力の波形を表現しようとする周波数解析の1つの手法であります。信号解析や画像処理や量子力学など多くの分野に応用されています。

フーリエ変換と比較して、周波数特性を求める際の時間領域情報を残すことができ、さらに、広い周波数領域の解析が可能であるといったようなメリットがあります。

ウェーブレット(Wavelet)変換には、連続ウェーブレット変換と離散ウェーブレット変換があります。

連続ウェーブレット変換は、1つのウェーブレット関数を基に、それをスケーリングしたり、シフトしたものを信号  $f(t)$  に乗算し、対象時間全体にわたる和を計算します。連続ウェーブレット変換の場合は、信号を無理やり連続ウェーブレット変換式に適用し計算するため、かなりの情報が失われ逆変換ができなくなります。そのため、実用上では、離散ウェーブレット変換がよく利用されます。

離散ウェーブレット変換は、連続ウェーブレットと比べ、2のべき乗をベースにスケールと位置を選択するものです。元信号を高周波成分と低周波成分に分解し、分解された低周波成分をまた高周波成分と低周波成分に分解するという処理を繰り返し行うことと等価です。

Maple は、離散ウェーブレットに関する計算機能を提供しております。直交ウェーブレット族(Daubechies ウェーブレット、Symlet、Coiflet、Battle-Lemarie ウェーブレット)および双直交ウェーブレット族(CDF ウェーブレット、双直交スプラインウェーブレット)をサポートしています。

Maple には、**DiscreteTransforms** パッケージにそのためのコマンド群が定まっています。それらのコマンドの処理機能や操作方法の基本をこちらの基礎編で紹介します。

- [DiscreteWaveletTransform](#) : 離散ウェーブレット変換
- [InverseDiscreteWaveletTransform](#) : 逆離散ウェーブレットの変換
- [WaveletCoefficients](#) : スケーリング係数およびウェーブレット係数の計算
- [WaveletPlot](#) : マザーウェーブレットおよびファザーウェーブレットの描画

## ウェーブレット(Wavelet)変換・逆変換

```
> restart
```

デフォルトで桁数を 20 桁に変更します：

```
> Digits := 20:
```

**DiscreteTransforms** パッケージをロードします：

```
> with(DiscreteTransforms)
```

```
[DiscreteWaveletTransform, FourierTransform, InverseDiscreteWaveletTransform,  
  InverseFourierTransform, WaveletCoefficients, WaveletPlot]
```

(2.1)

外部からノイズデータをインポートします：

```
> data := ImportMatrix("C:\\waveletvector.csv", datatype=float[8]);
```

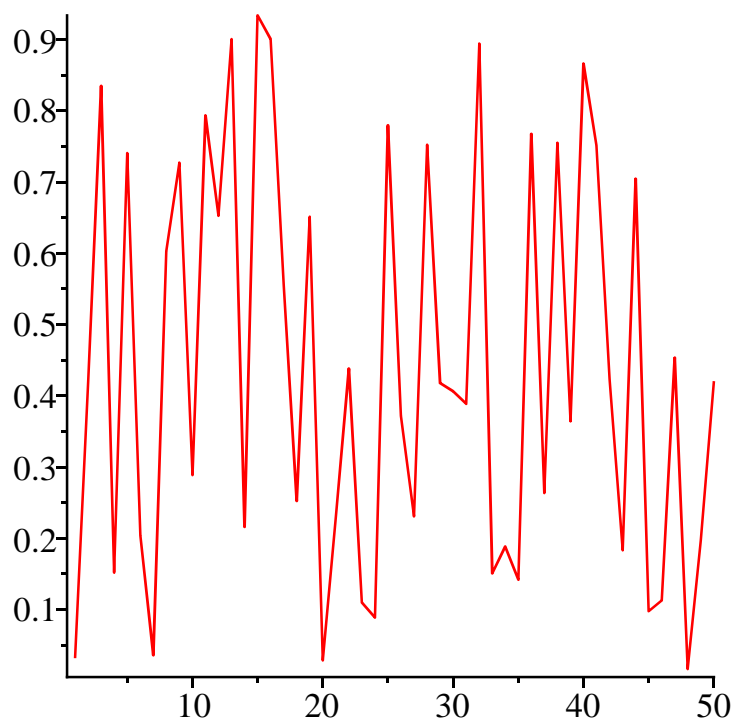
```
data := [ 50 x 2 Matrix  
         Data Type: float8  
         Storage: rectangular  
         Order: Fortran_order ]
```

(2.2)

上記データを描画します：

```
> with(plots):
```

```
> pointplot(data, style=line, color=red)
```



上記データを Daubechies ウェーブレットを用いて変換を行います：  
フィルタの長さを 4 にしています。  
ExT1 は、スケーリング係数、ExT2 はウェーブレット係数です。

```
> ExT1, ExT2 := DiscreteWaveletTransform(data[1..-1, 2], Daubechies, 4)
```

```
ExT1, ExT2 := 

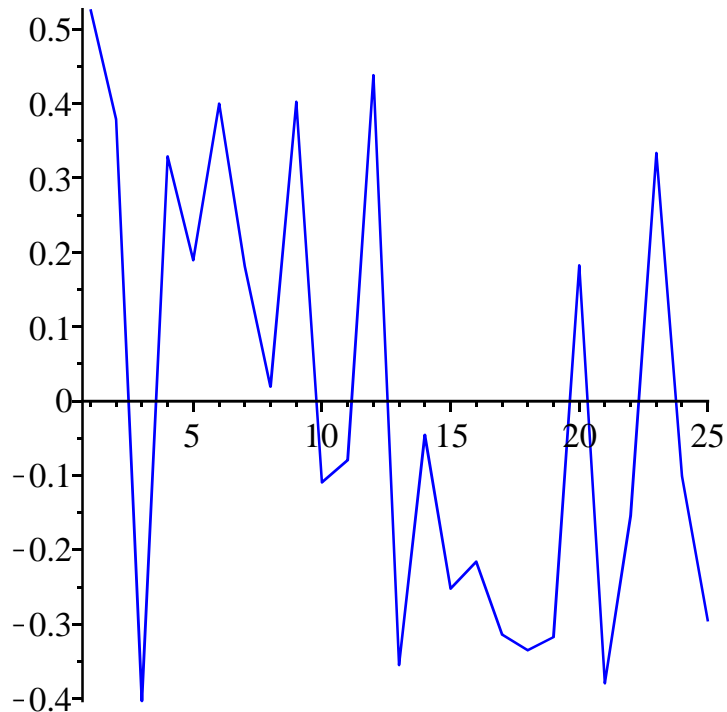
|   |                                        |   |   |                                        |   |
|---|----------------------------------------|---|---|----------------------------------------|---|
| [ | <i>1 .. 25 Vector<sub>column</sub></i> | , | [ | <i>1 .. 25 Vector<sub>column</sub></i> | ] |
|   | <i>Data Type: float<sub>8</sub></i>    |   | [ | <i>Data Type: float<sub>8</sub></i>    |   |
|   | <i>Storage: rectangular</i>            |   | [ | <i>Storage: rectangular</i>            |   |
|   | <i>Order: Fortran_order</i>            |   | [ | <i>Order: Fortran_order</i>            |   |


```

(2.3)

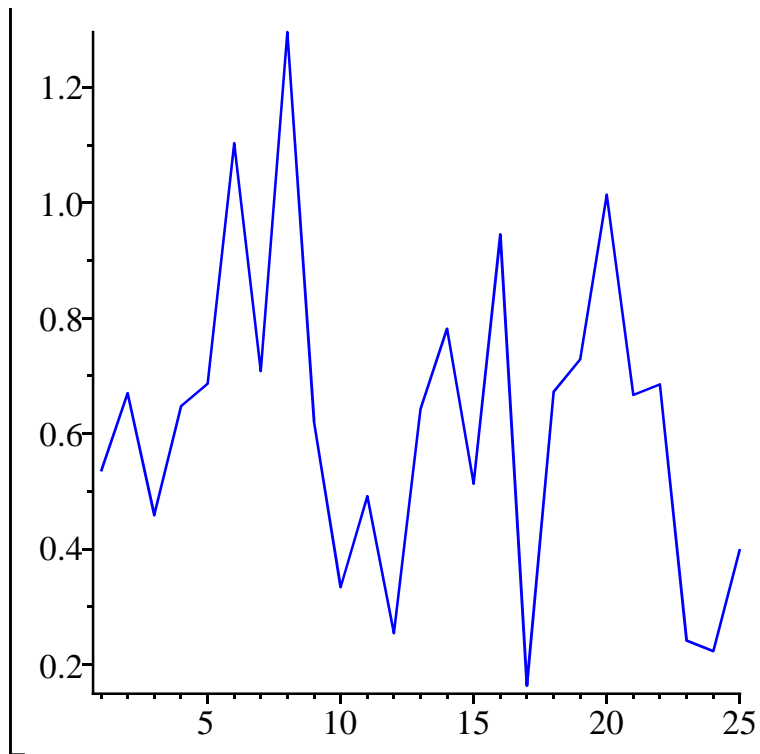
スケーリング係数のグラフを描画します：

```
> pointplot(Vector([seq(i, i = 1 .. 25)]), ExT1, style = line,  
color = blue)
```



ウェーブレット係数のグラフを描画します：

```
> pointplot(Vector([seq(i, i = 1 .. 25)]), ExT2, style = line,  
color = blue);
```

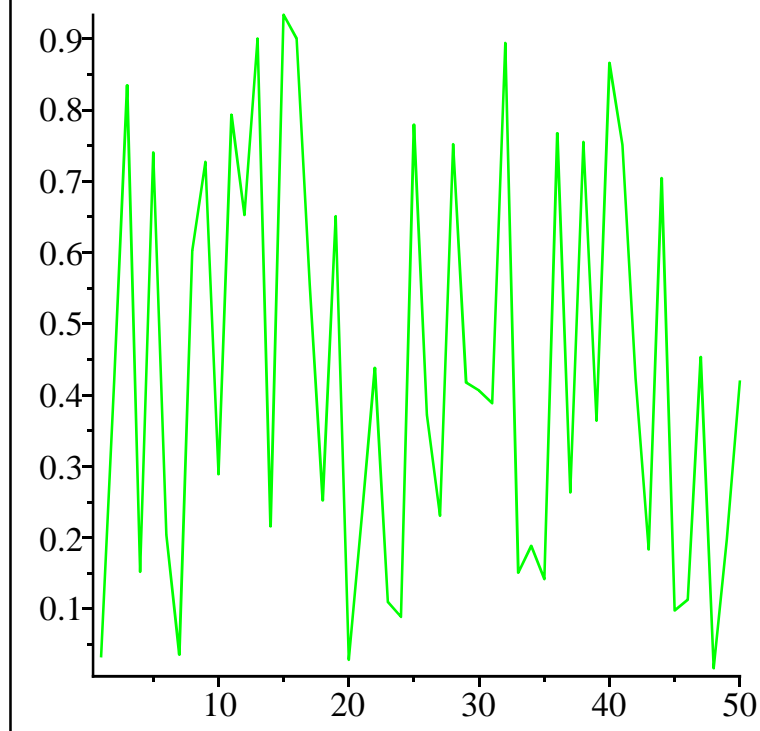


逆ウェーブレット変換を行います：

```
> IDWdata := InverseDiscreteWaveletTransform(ExT1, ExT2, Daubechies, 4) :
```

逆変換後のグラフを描画します：

```
> pointplot(Vector([seq(i, i = 1 .. 50)]), IDWdata, style =  
line, color = green)
```

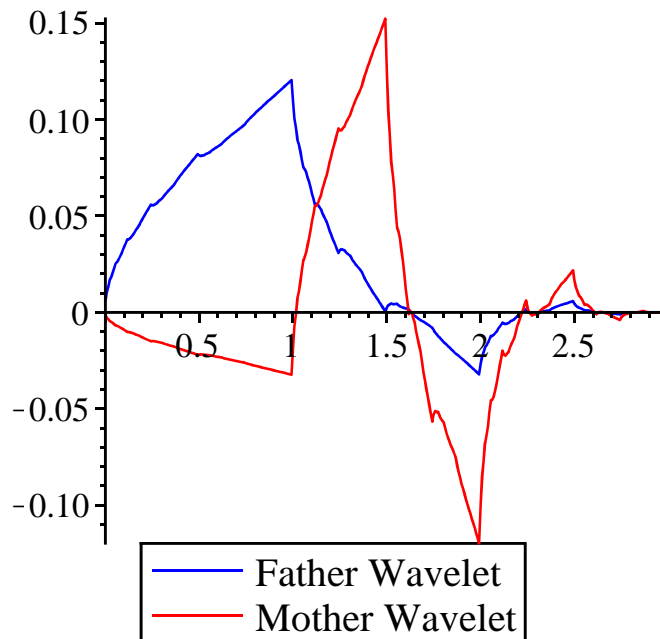


## マザーウェーブレット・ファザーウェーブレット

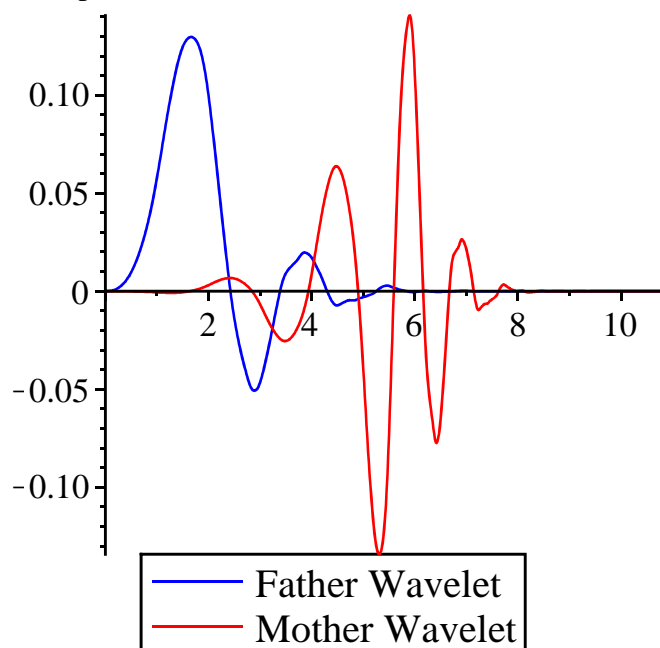
WaveletCoefficients コマンドにより、該当ウェーブレットとフィルタサージのスケーリング係数およびウェーブレット係数を算出します。  
さらに、その結果を WaveletPlot コマンドに適用させると、マザーウェーブレット、ファザーウェーブレットを計算し、描画を行います。

Daubechies 4 , Daubechies 12 , Daubechies 20 について、それぞれのマザーウェーブレット・ファザーウェーブレットを描画します：

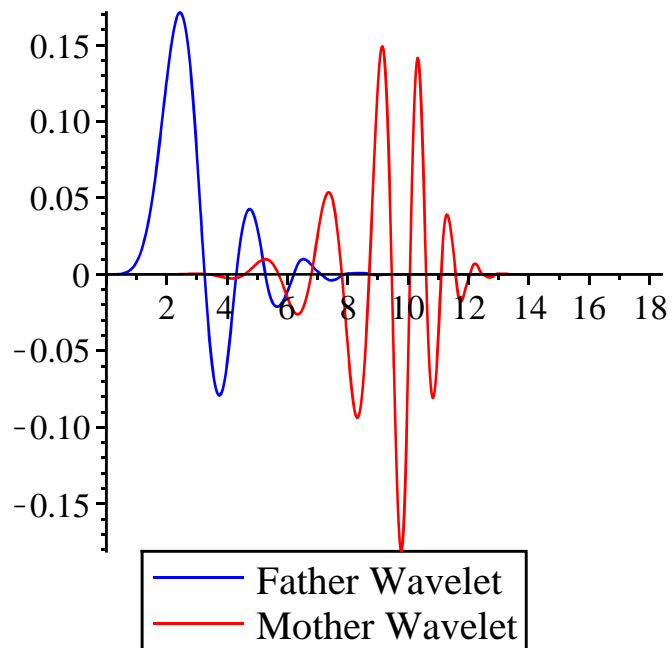
```
> WaveletPlot(WaveletCoefficients(Daubechies, 4))
```



```
> WaveletPlot(WaveletCoefficients(Daubechies, 12),  
minpoints = 500)
```



```
> WaveletPlot(WaveletCoefficients(Daubechies, 20),  
minpoints = 500)
```



## ▼ 簡単な画像変換例 :

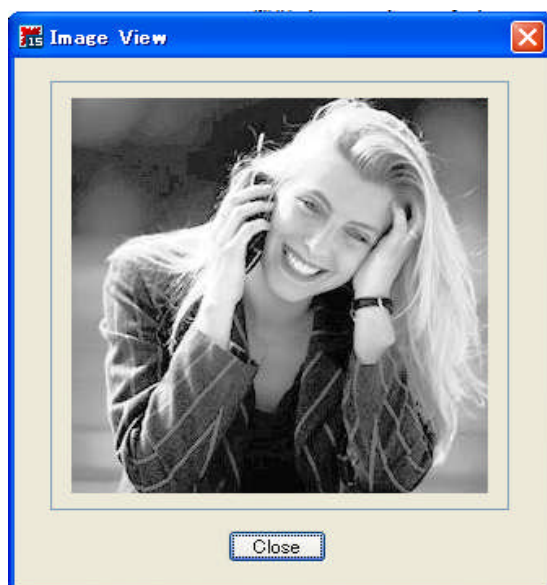
Maple の **ImageTools** 画像処理パッケージと合わせて簡単な応用例をご紹介します。

```
> restart
```

**ImageTools** パッケージおよび **DiscreteTransforms** 離散ウェーブレットをロードします :

```
> with(ImageTools) : with(DiscreteTransforms) :
```

イメージを読み込んで確認 :



```
> PreImg := Read(cat(kernelopts(datadir), "/images/phone.jpg")) :  
> View(PreImg)
```

上記データを離散ウェーブレット変換します：  
これにより、元の画像を低周波成分と高周波成分に分解します：

```
> ExImgHi, ExImgLo := DiscreteWaveletTransform(PreImg, 1, Daubechies, 20) :
```

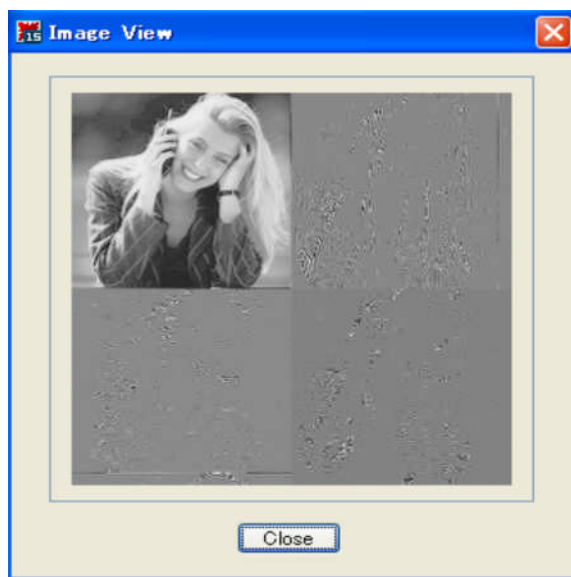
さらに上記分解結果について、離散ウェーブレット変換を行います：  
これにより、LL,HL,LH,LLに分解され、それらの係数が得られます：

```
> ExImgHiHi, ExImgHiLo, ExImgLoHi, ExImgLoLo  
:= DiscreteWaveletTransform(ExImgHi, 2, Daubechies, 20),  
DiscreteWaveletTransform(ExImgLo, 2, Daubechies, 20) :
```

上記処理により得られた各成分の係数を表示します：

```
> Show := <<FitIntensity(ExImgLoLo)|FitIntensity(ExImgLoHi), <FitIntensity(ExImgHiLo)  
|FitIntensity(ExImgHiHi)>> :
```

```
> View(Create(Show))
```



低周波	y 方向高周波
x 方向高周波	xy 方向高周波

このウェーブレット(Wavelet)変換～基礎編では、ウェーブレット(Wavelet)変換の定義や Maple が取り扱っている離散ウェーブレット変換の機能・簡単な事例に関して、解説しています。  
それらの機能や他の数式数値処理機能を組み合わせることで、Maple にてさらに様々な適用分野に活用していただけます。  
また、Maple 本体にウェーブレット(Wavelet)変換に関する応用例リファレンスも含まれており、適宜にご参照ください。