ウエーブレットと線形変換

Masato Shimura
JCD02773@nifty.ne.jp

平成 20 年 2 月 20 日

目 次

概 要

ウエーブレットは非定常や分散の変化する時系列に適応できる。配列を用いた多重解像度分解の計算手法を整理し、幾つかの経済時系列へ応用する。

はじめに

波型の解析では大小の *sin, cos* の組み合わせで波形をシュミレートするフーリエ変換が知られている。フーリエ変換は定常波には強いが、非定常や分散の変化する波の場合は複雑な計算を必要とした。

ウエーブレット小史　ウエーブレットはフランスの石油探査会社の技師 *Jean Morlet* が 1980 年代初頭に考案したとされる。モレはマルセイユの理論物理学者 *Grossmann* のところに持ち込み、マルセイユで理論的基礎が構築された。[1]

1985 年にフランス人の数学者 *Yves Meyer*（ドフィンヌ大）が同僚の物理学者にウエーブレットを紹介されて興味を持ち、マルセイユへ赴いて共同研究を行い、数学面の理論化を行った。1986 年にペンシルバニア大学でメイエの講演を聴いたコンピュータ画像を研究していたフランス人の大学院生 *Stephane Mallat* が、過去にも信号や画像の処理など工学分野で類似の手法が多く研究されていることを指摘し、メイエとの共同研究で多重解像度解析の理論が誕生した。[2]波の周辺には音楽家と同じ名前がよく現れる。

1　ハール · ウエーブレット

1.1　多重解像度分解 · カスケード分解

次のテーブルの *no* を 2 個ずつ、次のように計算する

足して 2 で割る	スケーリング係数	上段	*low pass filter*
引いて 2 で割る	ウエーブレット係数	下段 **bold**	*high pass filter*

更に *n*1 のスケーリング係数 (細字) のみを取り出して 2 個ずつを同様に組み合わせて計算する。*n*2, *n*3.. も同様に計算する。

このような単純な計算で波がカスケードに分解できる。下降サンプリングと言われる。

[1] この素敵な囁きを持つ言葉に従来からの同様の論考やアイデアも吸い寄せられて、短時間に理論が形成された。信号や画像処理系の利用が先行したので、用語や記述法も電気系か数学系の難解なものが多い

[2] このような単純なアルゴリズムが最近まで残っていたことが不思議である。カスケード分解は画像系の人達はメモリと計算速度に眼が行き、美しい構造を見逃していたようだ

*n*0(*data*)	*n*1	*n*2	*n*3
0	3.03265	5.0292	4.34192
6.0653	**-3.03265**		
7.3576	7.02575	**-1.99655**	
6.6939	**0.33185**		
5.4134	4.7588	3.65465	**0.687275**
4.1042	**0.6546**		
2.9872	2.5505	**1.10415**	
2.1138	**0.4367**		

```
  2^i.12
1 2 4 8 16 32 64 128 256 512 1024 2048
```

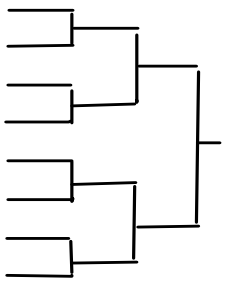

図 1: トーナメント表

スケーリング関数の部分のみを取り出してグラフにすると波の分解されている様子が分かる。

これは多重解像度解析と呼ばれる離散データを順次低周波から高周波への成分で表したデータの線形結合に分解する手法である。

```
  cascade_hwt  K0
+-----------------------------------------------------------+
|0 6.0653 7.3576 6.6939 5.4134 4.1042 2.9872 2.1138         |
+-----------------------------------------------------------+
|3.03265 _3.03265 7.02575 0.33185 4.7588 0.6546 2.5505 0.4367|
+-----------------------------------------------------------+
```

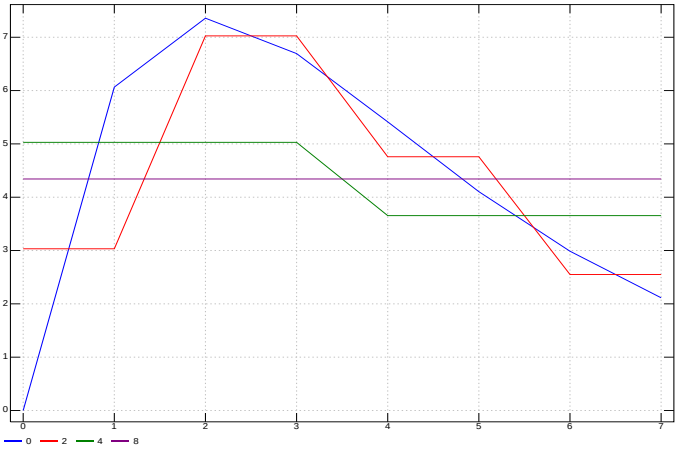

図 2: ハールの多重解像度分解

```
|5.0292 _1.99655 3.65465 1.10415                                |
+---------------------------------------------------------------+
|4.34192 0.687275                                               |
+---------------------------------------------------------------+
```

トーナメント　囲碁や将棋の NHK 杯や高校野球のトーナメントは一回戦を終了した時点で 32 名（校）が残る。以降の隣同士が勝ち残りを争う手法はウエーブレットの多重解像度解析と同様である。経済の時系列データ等が 2^n になることは希であり、残ったデータの取り扱いには工夫が要る。

1.2　高速ハール · ウエーブレット変換

高速と唱っているが、単なる線形計算である。

先の足して 2 で割る、引いて 2 で割るがハール · ウエーブレットのコアである。*Alfred Haar* によって 1909 年に提案された。

次のマトリクスに辿り着く迄の道筋を簡単にフォローする。奇数行 (a) はスケーリング関数を、偶数行 (c) はウエーブレット関数を計算する。

$$\begin{pmatrix} a_0 \\ c_0 \\ a_1 \\ c_1 \\ a_2 \\ c_2 \\ a_3 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & -0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & -0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \end{pmatrix} \times \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix}$$

```
   mat_hwt_sub K0
0.5  0.5   0    0   0    0   0    0
0.5 _0.5   0    0   0    0   0    0
  0    0 0.5  0.5   0    0   0    0
  0    0 0.5 _0.5   0    0   0    0
  0    0   0    0 0.5  0.5   0    0
  0    0   0    0 0.5 _0.5   0    0
  0    0   0    0   0    0 0.5  0.5
  0    0   0    0   0    0 0.5 _0.5
```

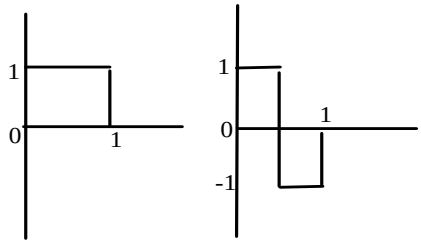

図 3: mother wavletϕ,scaling ψ

ϕ Scaling	$\phi(t) = \begin{pmatrix} 1 & 0 \le t < 1 \\ 0 & otherwise \end{pmatrix}$	ψ Wavelet	$\phi(t) = \begin{pmatrix} 1 & 0 \le t < 1/2 \\ -1 & 1/2 \le t < 1 \\ 0 & otherwise \end{pmatrix}$

$0\phi_{(0,0.5)} + 6.0653\phi_{(0.5,1)} = \frac{0 + 6.0653}{2}\phi_{(0,1)} + \frac{0 - 6.6053}{2}\psi_{(0,1)} = 3.03265\phi_{(0,1)} - 3.03265\psi_{(0,1)}$

$H_2 = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix}$ の形をデータ (2^n に合わせる。) の個数に展開すれば最初のマトリクスとなる。カスケードの計算過程で n が $\frac{1}{2}$ ずつ縮小していくのでマトリクスもその都度縮小形に作り変える。

1.3 ハール・ウエーブレットの逆変換

ウエーブレットは逆変換が可能である。スケーリングとウエーブレットの値をサンドイッチにして小さなマトリクスから逆に組み上げていく。

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix} = \begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{matrix} \times \begin{pmatrix} a_0 \\ c_1 \\ a_2 \\ c_3 \\ a_4 \\ c_5 \\ a_6 \\ c_7 \end{pmatrix}$$

0	$\begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix} \times \begin{pmatrix} a_0 \\ c_0 \end{pmatrix}$	a1=.4.34192 0.687275 (1 1,:1 _1) +/ . * a1 1 1 1 _1 (1 1,:1 _1) +/ . * a1 5.0292 3.65465

```
   3 reverse_hwt cascade_hwt K0
5.0292 3.65465
   2 reverse_hwt cascade_hwt K0
3.03265 7.02575 4.7588 2.5505
   1 reverse_hwt cascade_hwt K0
0 6.0653 7.3576 6.6939 5.4134 4.1042 2.9872 2.1138
```

2 ドビッシー・ウエーブレット

34

[3]響きの良いフランス語読みとする。

[4]*Ingrid Daubechies* ベルギー生まれの女性研究者。ベルギーで ph.D を取り、ベル研究所などを経て現在プリンストン大学。一時、マルセイユグループにいたようだ。

2.1 ドビッシー基底

ドビッシは 1987 年にカスケードアルゴリズムでのコンパクト台を持つウエーブレット基底を考案した。

2.1.1 D4

N2(D4) の場合。なお、(N1(D2) は *Haar* 基底と同じである。

Scaling 規格化の場合。規格化を行わない場合は $\frac{1+\sqrt{3}}{4}$ の様に $\sqrt{2}$ を落とす。双方とも用いられているようだ。

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}$$
$$h_0 = \frac{3+\sqrt{3}}{4\sqrt{2}}$$
$$h_0 = \frac{3-\sqrt{3}}{4\sqrt{2}}$$
$$h_0 = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

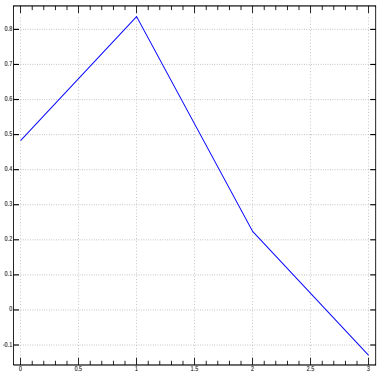

図 4: mother wavletϕ

Wavelet $g_0 = h_3$

$g_1 = -h_2$

$g_2 = h_1$

$g_3 = -h_0$

```
    dp_40 ''
0.482963 0.836516 0.224144 _0.12941
    dp_41 ''
0.683013 1.18301 0.316987 _0.183013
```

2.2 高速ドビッシウエーブレット変換とエッジ問題

D4 の係数を次式によりマトリクスにする。$N = 2^k$ である。最後の 2 行がマトリクスをはみ出す。データを周期的と見なして回り込みを許すか最後の 2 行のデータを *AR*（自己回帰）などで推計してこの部分の計算を工夫する必要がある。

$$\begin{pmatrix} a_0^{n-1} \\ a_1^{n-1} \\ a_2^{n-1} \\ a_3^{n-1} \\ a_4^{n-1} \\ a_5^{n-1} \\ a_6^{n-1} \\ a_7^{n-1} \end{pmatrix} = \frac{1}{2} \left(\begin{array}{cccccccc|cc} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 & & \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 & & \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & & \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \end{array}\right) \times \begin{pmatrix} a_0^n \\ a_1^n \\ a_2^n \\ a_3^n \\ a_4^n \\ a_5^n \\ a_6^n \\ a_7^n \end{pmatrix}$$

[5]

```
  6j2 ": mat_dwt4_sub K0 NB. 配列部分　回り込み許容
 0.48  0.84  0.22 _0.13  0.00  0.00  0.00  0.00
_0.13 _0.22  0.84 _0.48  0.00  0.00  0.00  0.00
 0.00  0.00  0.48  0.84  0.22 _0.13  0.00  0.00
 0.00  0.00 _0.13 _0.22  0.84 _0.48  0.00  0.00
 0.00  0.00  0.00  0.00  0.48  0.84  0.22 _0.13
 0.00  0.00  0.00  0.00 _0.13 _0.22  0.84 _0.48
 0.22 _0.13  0.00  0.00  0.00  0.00  0.48  0.84
 0.84 _0.48  0.00  0.00  0.00  0.00 _0.13 _0.22
```

2.3 $D(4)$ の逆変換

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix} = \left(\begin{array}{cc|cccccccc} h_2 & g_2 & h_0 & g_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ h_3 & g_3 & h_1 & g_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & h_2 & g_2 & h_0 & g_0 & 0 & 0 & 0 & 0 \\ & & h_3 & g_3 & h_1 & g_1 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & h_2 & g_2 & h_0 & g_0 & 0 & 0 \\ & & 0 & 0 & h_3 & g_3 & h_1 & g_1 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & h_2 & g_2 & h_0 & g_0 \\ & & 0 & 0 & 0 & 0 & h_3 & g_3 & h_1 & g_1 \end{array}\right) \times \begin{pmatrix} a_0 \\ c_0 \\ a_1 \\ c_1 \\ a_2 \\ c_2 \\ a_3 \\ c_3 \end{pmatrix}$$

[5] トーナメントの階層と同じで最後の 2 段に特に効いている

3　景気動向指数への応用

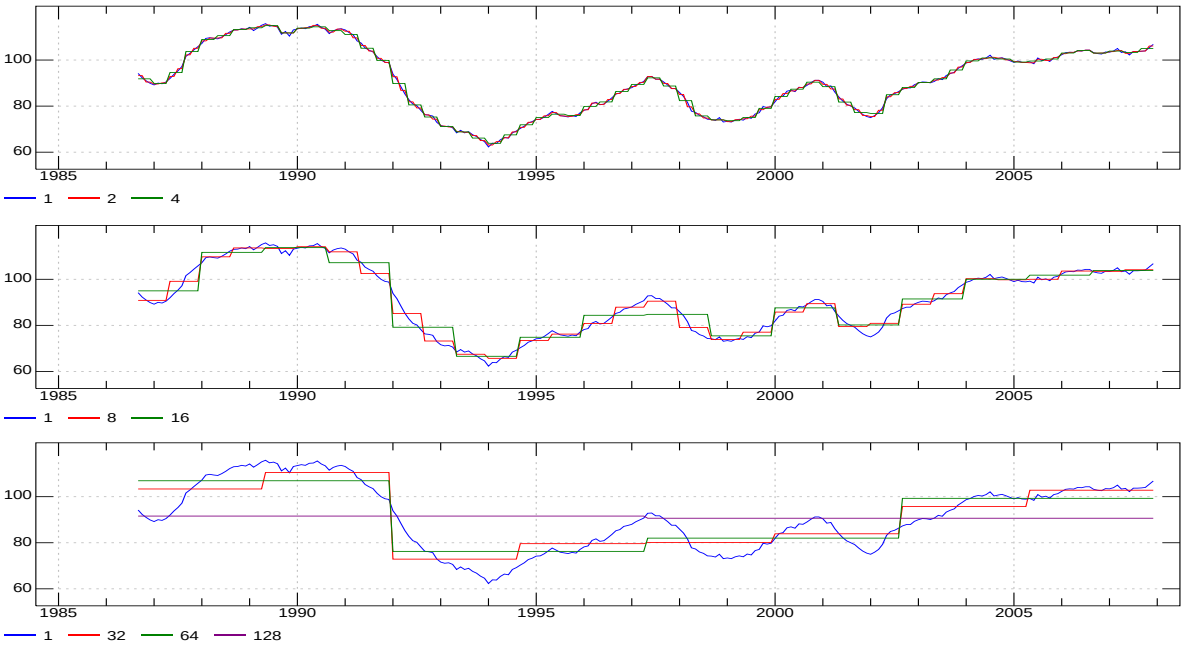

図 5: C17 電力消費量　 Haar

4　関数一覧

classify	main	sub
file	*wavelet_0.ijs*	
haar	cascade_hwt	pick_hwt_sub reverse_hwt_sub mat_hwt_sub
daubechies	cascade_dwt40 NB. norm cascade_dwt41 NB. non norm	mat_dwt4_sub0 dp4_0 NB. norm mat_dwt4_sub1 dp4_1 NB.non norm

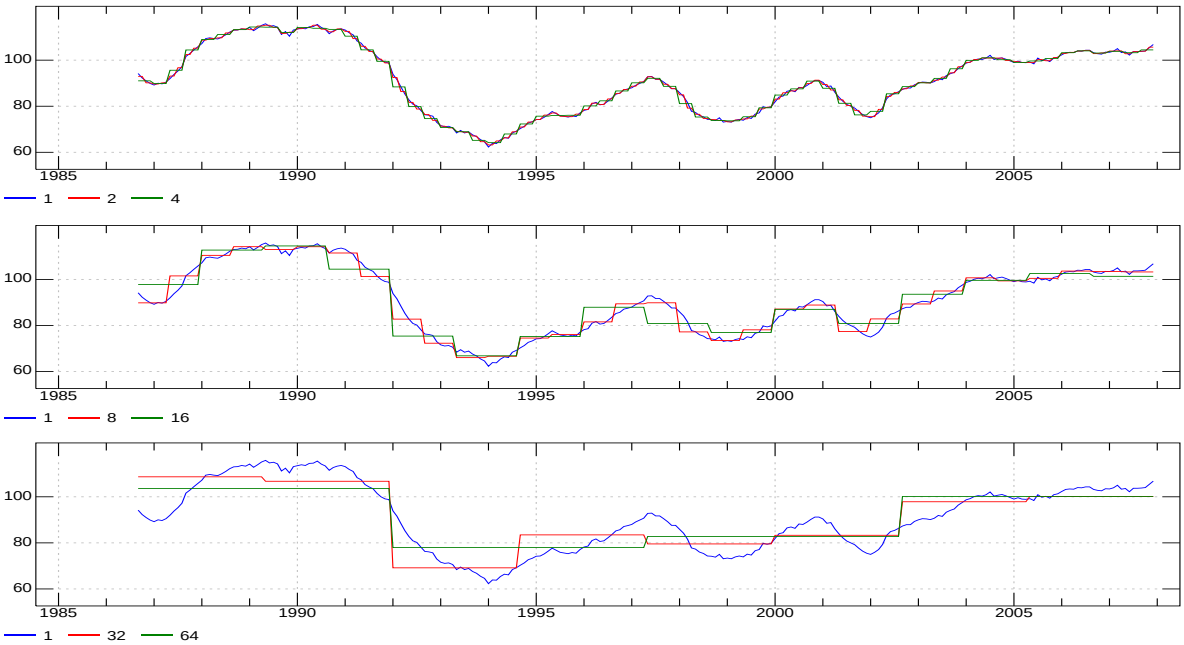

図 6: C17 電力消費量　dw4

common		pick_scale pick_wavelet norm0
plot	plot_cascade plot_cascade2	
sna	wavelet_sna plot_cascade3	mk_xaxis_wavelet read_trend_data

5　Reference

Yves Nievergelt, 松本 · 雛元 · 茂呂訳「ウエーブレット変換の基礎」森北出版　2004

金谷健一「これなら分かる応用数学教室」共立出版　2003

A　ノルム化

A.1　haar

$$h = \begin{matrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{matrix}$$

を用いる。

A.2　ドビッシー D4

規格化の場合。

$$h_0 = \frac{1 + \sqrt{3}}{4\sqrt{2}}$$

$$h_0 = \frac{3 + \sqrt{3}}{4\sqrt{2}}$$

$$h_0 = \frac{3 - \sqrt{3}}{4\sqrt{2}}$$

$$h_0 = \frac{1 - \sqrt{3}}{4\sqrt{2}}$$

B　ドビッシー $D4$ のエッジ

B.1　厳正な補正

ドビッシーの $D(4)$ では 2 個エッジが必要となる。経済時系列では最新データが最後にくることが多いのでエッジを左上に持ってきた方がよい。

ここに 2 個のデータを何段にも持って来るには $2, 6, 14, 30, 62..$ のデータを必要とする。

$$\begin{pmatrix} a_0^{n-1} \\ a_1^{n-1} \\ a_2^{n-1} \\ a_3^{n-1} \\ a_4^{n-1} \\ a_5^{n-1} \\ a_6^{n-1} \\ a_7^{n-1} \end{pmatrix} = \left(\begin{array}{cc|cccccccc} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \end{array}\right) \times \begin{pmatrix} a_0^n \\ a_1^n \\ a_2^n \\ a_3^n \\ a_4^n \\ a_5^n \\ a_6^n \\ a_7^n \end{pmatrix}$$

B.2　回り込みのシンプルな補正

経済時系列では最新（右下）が重要なので、最初の方で補正する。[6]

*mat_dwt*4_*sub*1

$$\begin{pmatrix} h_2 & h_3 & 0 & 0 & 0 & 0 & h_0 & h_1 \\ g_2 & g_3 & 0 & 0 & 0 & 0 & g_0 & g_1 \\ h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \end{pmatrix}$$

C　D(4) の計算

Kristian Sanberg(Dept. of Applied Mathematics,Univ. Cololado) による計算手順. 回り込みを利用している。

$c = (c(0), c(1), c(2), c(3)) = (\frac{1+\sqrt{3}}{4\sqrt{2}}, \frac{3+\sqrt{3}}{4\sqrt{2}}, \frac{3-\sqrt{3}}{4\sqrt{2}}, \frac{1-\sqrt{3}}{4\sqrt{2}})$

(*as a low pass filter*)

$d = (d(0), d(1), d(2), d(3)) = (c(3), c(3), c(1), c(0))$

(*as a high pass filter*)

$\frac{1}{\sqrt{2}}$ *is a normalization facter*

Consider $f = (f(0), f(1), \cdots, f(7)) = (2, 5, 8, 9, 7, 4, -1, 1)$

$Step1$:

$\tilde{f} = (f(6), f(7), f(0), f(1), f(2), f(3), f(4), f(5), f(6), f(7)) = (-1, 1, 2, 5, 8, 9, 7, 4, -1, 1)$

$$f_1 = \begin{pmatrix} c(0)f(6) & +c(1)f(7) & +c(2)f(0) & c(3)f(1) \\ c(0)f(0) & +c(1)f(1) & +c(2)f(2) & c(3)f(3) \\ c(0)f(2) & +c(1)f(3) & +c(2)f(4) & c(3)f(5) \\ c(0)f(4) & +c(1)f(5) & +c(2)f(6) & c(3)f(7) \\ d(0)f(6) & +d(1)f(7) & +d(2)f(0) & d(3)f(1) \\ d(0)f(0) & +d(1)f(1) & +d(2)f(2) & d(3)f(3) \\ d(0)f(2) & +d(1)f(3) & +d(2)f(4) & d(3)f(5) \\ d(0)f(4) & +d(1)f(5) & +d(2)f(6) & d(3)f(7) \end{pmatrix}$$

$$= \texttt{0.154794 5.77697 12.4437 6.37325 \_0.836516 0.965926 0.871191 \_3.12192}$$

[6]一見うまく行ってそうだがグラフ上では 1 拍ズレる

$Step2:$

$\tilde{f}_1 = (f_1(2), f_1(3), f_1(0), f_1(1), f_1(2), f_1(3), f_1(4), f_1(5), f_1(6), f_1(7))$

= 12.4437 6.37325 0.154794 5.77697 12.4437 6.37325 _0.836516 0.965926 0.871191 _3.12192

$$f_2 = \begin{pmatrix} c(0)f_1(2) & +c(1)f_1(3) & +c(2)f_1(0) & c(3)f_1(1) \\ c(0)f_1(0) & +c(1)f_1(1) & +c(2)f_1(2) & c(3)f_1(3) \\ d(0)f_1(2) & +d(1)f_1(3) & +d(2)f_1(0) & d(3)f_1(1) \\ d(0)f_1(0) & +d(1)f_1(1) & +d(2)f_1(2) & d(3)f_1(3) \\ & f_1(4) & & \\ & f_1(5) & & \\ & f_1(6) & & \\ & f_1(7) & & \end{pmatrix}$$

= 10.6283 6.87172 _5.69944 6.01642 _0.836516 0.965926 0.871191 _3.12192

$Step3:$

$\tilde{f}_2 = (f_2(0), f_2(1), f_2(0), f_2(1), f_2(2), f_2(3), f_2(4), f_2(5), f_2(6), f_2(7))$

= 10.6283 6.87172 10.6283 6.87172 _5.69944 6.01642 _0.836516 0.965926 0.871191 _3.12192

$$f_3 = \begin{pmatrix} c(0)f_2(0) & +c(1)f_2(1) & +c(2)f_2(0) & c(3)f_2(1) \\ d(0)f_2(0) & +d(1)f_2(1) & +d(2)f_2(0) & d(3)f_2(1) \\ & f_2(2) & & \\ & f_2(3) & & \\ & f_2(4) & & \\ & f_2(5) & & \\ & f_2(6) & & \\ & f_2(7) & & \end{pmatrix}$$

12.3744 2.6563 _5.69944 6.01642 _0.836516 0.965926 0.871191 _3.12192

cas0_dwt4 S0

```
+-----------------------------------------------------------------------+
|2 5 8 9 7 4 _1 1                                                       |
+-----------------------------------------------------------------------+
|0.154794 5.77697 12.4437 6.37325 _0.836516 0.965926 0.871191 _3.12192|
+-----------------------------------------------------------------------+
|10.6283 6.87172 _5.69944 6.01642                                       |
+-----------------------------------------------------------------------+
|12.3744 2.6563                                                         |
+-----------------------------------------------------------------------+
```

D　大型粗行列の計算

```
   7j4 ":  dp4_0 mat_dwt4_sub0 K0
 0.4830 0.8365 0.2241_0.1294 0.0000 0.0000 0.0000 0.0000
_0.1294_0.2241 0.8365_0.4830 0.0000 0.0000 0.0000 0.0000
 0.0000 0.0000 0.4830 0.8365 0.2241_0.1294 0.0000 0.0000
 0.0000 0.0000_0.1294_0.2241 0.8365_0.4830 0.0000 0.0000
 0.0000 0.0000 0.0000 0.0000 0.4830 0.8365 0.2241_0.1294
 0.0000 0.0000 0.0000 0.0000_0.1294_0.2241 0.8365_0.4830
 0.2241_0.1294 0.0000 0.0000 0.0000 0.0000 0.4830 0.8365
 0.8365_0.4830 0.0000 0.0000 0.0000 0.0000_0.1294_0.2241
```

大型粗行列にはスパースアレー$. がサポートされている。0 以外の数値のアドレスを記憶して計算する事でメモリ節約となる。

```
  $.  dp4_0 mat_dwt4_sub0 K0
0 0 |  0.482963
0 1 |  0.836516
0 2 |  0.224144
0 3 |  _0.12941
1 0 |  _0.12941
1 1 | _0.224144
1 2 |  0.836516
1 3 | _0.482963
2 2 |  0.482963
2 3 |  0.836516
2 4 |  0.224144
2 5 |  _0.12941
```

E　Script

E.1　Haar Wavelet

NB. 内積 (+/ . *) でハールマトリクスとデータを計算する。
2 個になるまで反復

```
cascade_hwt=: 3 : 0
NB. cascade Haar wavelet
TIME_IND=:  I. (2^i. # y) e. # y NB. index for repeat
ANS=.< TMP=. y
for_CTR. i.  TIME_IND  do.
```

```
MAT=. mat_hhwt_sub TMP
N0=. MAT +/ . * TMP
ANS=. ANS,<N0
TMP=. pick_hwt_sub N0 NB. pick upper of sandwitch
end.
,. ANS
)
```

NB. スケーリング部分の値のみを取り出す

```
pick_hwt_sub=: 3 : '(-.*(2|i.# y))# y' NB. pick odd is scaling
```

NB. マトリクスへの展開。|.("0 1) と回転させて実数 (0.5,-0.5 etc) を所定位置へ配置

```
mat_hwt_sub=: 3 : 0
NB. mat for highspeed Haar wavelet
INDX =.-: # y NB. half of number
TMP0=. ;("2),.INDX # < 2 2 $ 0.5 0.5 0.5 _0.5
TMP1=. TMP0 ,. (;(# TMP0),((# y)-2))$0 NB. add 0 part
IND3=.- ;2 # L:0 {@> INDX{. +: i. # y  NB. make rotate index
IND3 |. ("0 1) TMP1
)
```

NB. スケーリングの値を取り出し、グラフ用に Expand。汎用に副詞（1：0) としている。

```
plot_cascade=: 1 : 0
TMP0=: u y
IND=: {@> }.2^i.  # TMP0
TMP1=. (-.&* (L:0) 2 | L:0 i. L:0 , # L:0 }.TMP0) # L:0 }. TMP0
DAT=.;("1)({. TMP0),IND # L:0 TMP1
pd 'reset'
pd 'keypos open bottom'
pd 'keystyle open horizontal'
pd 'key ',": 0,;IND
pd DAT
pd 'show'
)
```