Mathematica Activities

Off[General::spell]

Packages needed:

<<waves.m
<<Statistics`NormalDistribution`

Filters corresponding to Daubechies wavelets

Daubechies family of compactly supported wavelets, DAUB #n,
is the orthonormal basis in L^2(R) with properties:
(1) support of psi(x) is the interval [0,2 n-1];
(2) 0=int_R psi(x) dx = int_R x psi(x) dx = ... =
int_R x^(n-1) psi(x) dx, and
(3) psi is in C^{\alpha}, alpha={0.5-epsilon, 0.915, 1.275, 1.596,
1.888, 2.158,...} for n=2,3,4,5,6,...
The Haar wavelet is DAUB #1.

d1={0.7071067811865475, 0.7071067811865475};

d2={0.482962913145, 0.836516303738,
0.224143868042, -0.129409522551};

d4={0.230377813309, 0.714846570553, 0.630880767930,
-0.027983769417,-0.187034811719, 0.030841381836
0.032883011667 -0.010597401785};

d6={0.1115407433501098, 0.4946238903984543,
0.7511339080210969, 0.3152503517091975,
-0.2262646939654413,-0.1297668675672621,

```
   0.0975016055873234, 0.0275228655303057,
  -0.0315820393174862, 0.0005538422011614892,
   0.004777257510945529, -0.001077301085308485};
d10={0.026670057901, 0.188176800078, 0.527201188932,
   0.688459039454, 0.281172343661,-0.249846424327,
   -0.195946274377, 0.127369340336, 0.093057364604,
   -0.071394147166,-0.029457536822, 0.033212674059,
   0.003606553567,-0.010733175483, 0.001395351747,
   0.001992405295,-0.000685856695,-0.000116466855,
   0.000093588670,-0.000013264203};
```

Ploting the data set

```
DataPlot[li_List]:=
Plot[ li[[ Floor[ Length[li] x ] + 1 ]],
                    {x,0,0.99999}]
```

Thresholding

Hard Thresholding

```
ThresHard[list_List,thresh_]:=Module[{},
Table[ If[ Abs[list[[i]]] < thresh , 0 , list[[i]]] ,
                    {i,1,Length[list]}] ]
```

Soft Thresholding

```
ThresSoft[list_List, thresh_]:=
     Module[{ll=Length[list]},
Table[ If  [Abs[ list[[i]]] - thresh > 0,
  Sign[list[[i]]] ( Abs[list[[i]]] - thresh), 0],
  {i, 1,ll} ]
       ];
```

Universal Thresholding

```
ThresUniv[ list_List]:=
 Module[{ll=Length[list], s, unithresh},
   s=N[Sqrt[1/(ll-1) Apply[Plus,
      (list - Apply[Plus, list]/ll)^2 ] ]];
   unithresh=N[s Sqrt[ 2 Log[ll]] ];
  Table[ If[Abs[ list[[i]] ] - unithresh >0,
 Sign[ list[[i]] ] (Abs[list[[i]]] - unithresh),0],
      {i,1, ll} ]
];
```

Trimming

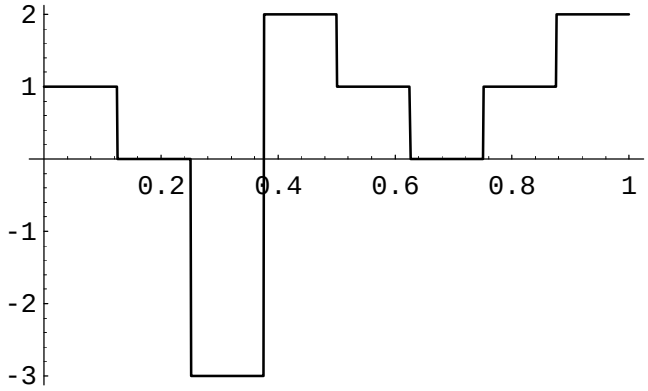
```
Trimm[list_List, lam_]:=
Table[ Abs[list[[i]]] Tanh[ lam list[[i]] ],
                {i,1, Length[list]}
    ];
```

Examples

Example from the tutorial, page 6.

```
example={1,0,-3,2,1,0,1,2};
```

DataPlot[example]



-Graphics-

ed1=WT[example, d1]

{0.7071067811865475, -3.53553, 0.7071067811865475, -0.707107, 1., -1., -1.41421,

1.41421}

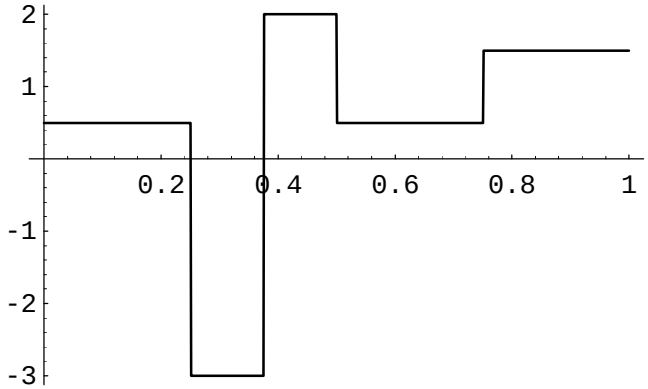
et1=ThresHard[ed1, 0.9]

{0, -3.53553, 0, 0, 1., -1., -1.41421, 1.41421}

er1=WR[et1, d1]

{0.5, 0.5, -3., 2., 0.5, 0.5, 1.5, 1.5}

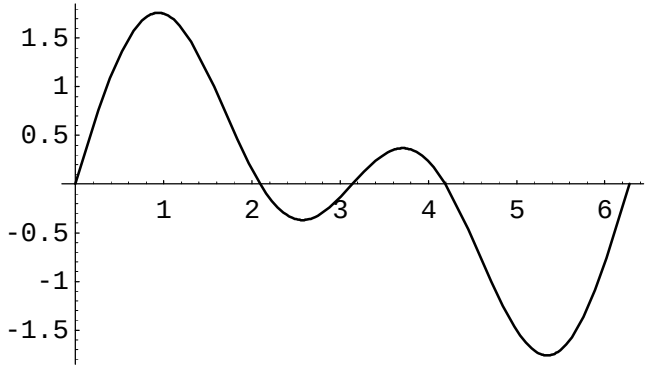
DataPlot[er1]



-Graphics-

Function Sin[x]+Sin[2 x]: Denoising by different wavelet bases and thresholding policies.

Plot[Sin[x] + Sin[2 x], {x, 0, 2 Pi}]



-Graphics-

RandomSeed[125];

li=Table[N[Sin[x]+Sin[2 x]+Random[
NormalDistribution[0, 0.4]]],
{x, 2 Pi/1024, 2 Pi, 2 Pi/1024}];

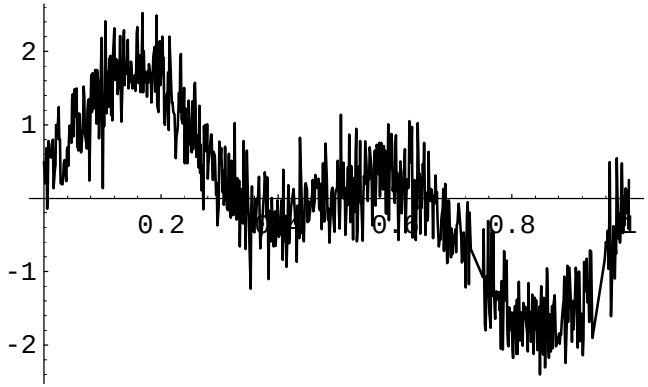
Short[li, 3]

{0.489125, 0.759563, 0.203772, 0.687917, -0.553575, -0.145796, 0.11052, 0.782839,

-0.113925, 0.607707, 0.577896, 0.550342, 0.408077, 0.729907, <<1003>>, 0.0199374,

0.134395, -0.208486, -0.163896, 0.111371, -0.415866, 0.246625}

DataPlot[li]



-Graphics-

lid1=WT[li, d1];

Short[lid1,3]

{-0.191229, -0.342343, -0.288343, -0.475401, -0.510271, 0.0194839, -0.227568,

-0.071192, -0.323387, -0.0477633, -0.417172, -0.0400722, -0.0278623, <<10>>7,

<<1003>>, 3.28927, 4.3442, -3.79269, 14.0525, 14.5291, 20.6674, 0.0359969}

lit1=ThresHard[lid1, 1.2];

Short[lit1, 3]

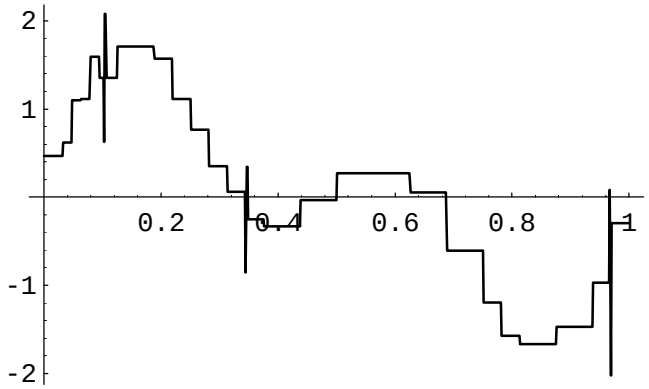
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, <<978>>, 2.08766, 3.70591, -1.67585, 0, 3.73001, 1.59364, -4.72405, -4.12402,

3.28927, 4.3442, -3.79269, 14.0525, 14.5291, 20.6674, 0}

lin1=WR[lit1, d1];

DataPlot[lin1]



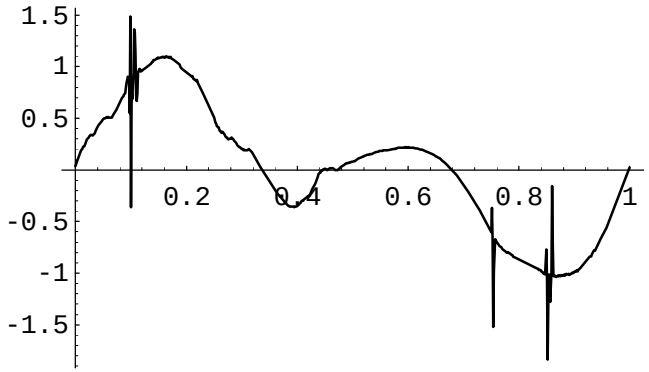
-Graphics-

lid4=WT[li, d4];

lit4=ThresHard[lid4,1.2];

lin4=WR[lit4, d4];

DataPlot[lin4]

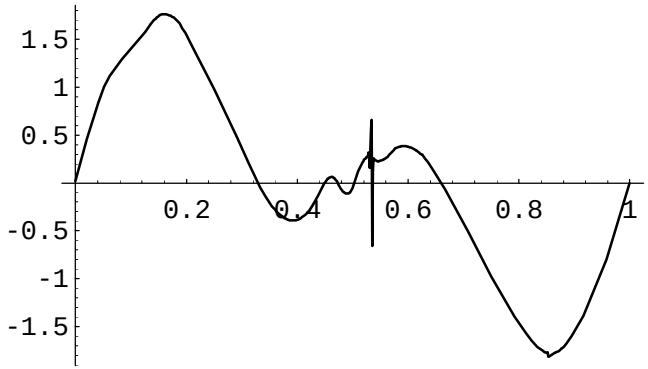


-Graphics-

Different thresholding policies and DAUB #6.

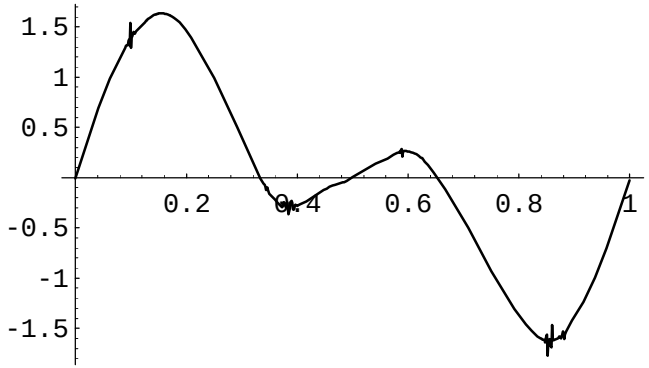
lid6=WT[li, d6];

lith6=ThresHard[lid6, 1.2];
linh6=WR[lith6, d6];
DataPlot[linh6]



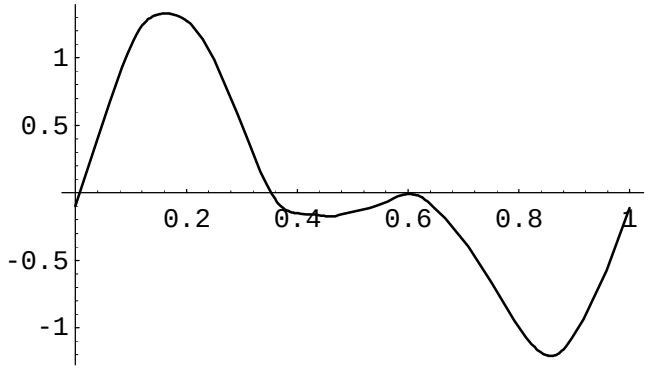
-Graphics-

```
lits6=ThresSoft[lid6, 1];
lins6=WR[ lits6, d6];
DataPlot[lins6]
```



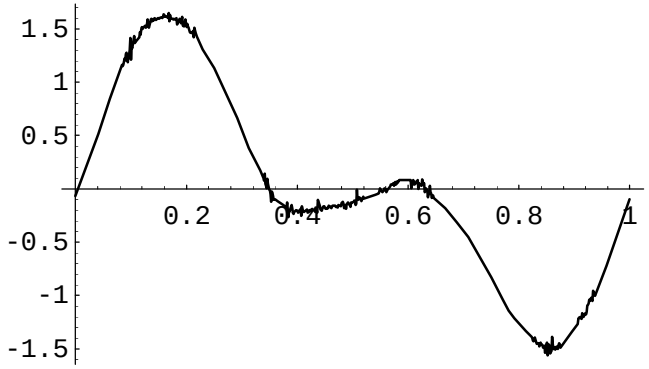
-Graphics-

```
litu6=ThresUniv[lid6];
linu6=WR[ litu6, d6];
DataPlot[linu6]
```



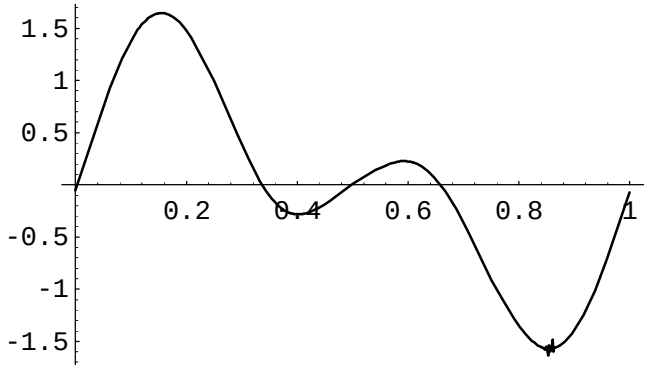
-Graphics-

```
litt6=Trimm[lid6, 0.1];
lint6=WR[litt6, d6];
DataPlot[lint6]
```



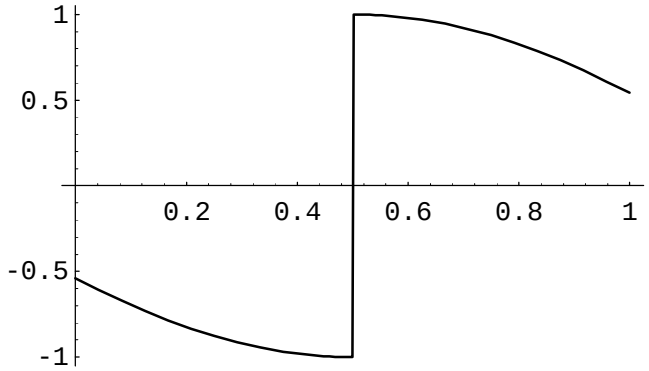
-Graphics-

lid10=WT[li, d10];
lit10=ThresSoft[lid10, 1.2];
lin10=WR[lit10, d10];
DataPlot[lin10]



-Graphics-

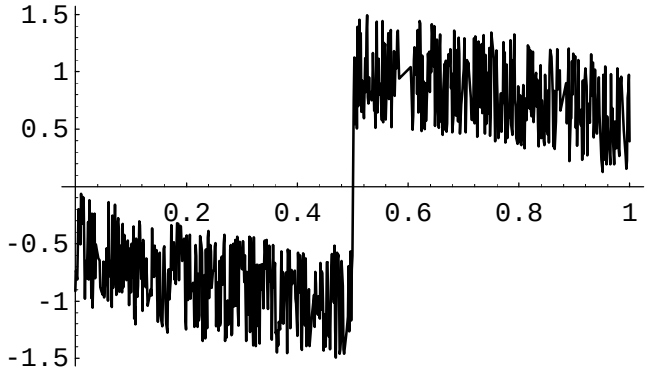
aa=Table[N[Sign[x] Cos[x]],{x,-1, 511/512, 1/512}];
DataPlot[aa]



-Graphics-

bb=aa+Table[(Random[]-0.5),{1024}];

DataPlot[bb]



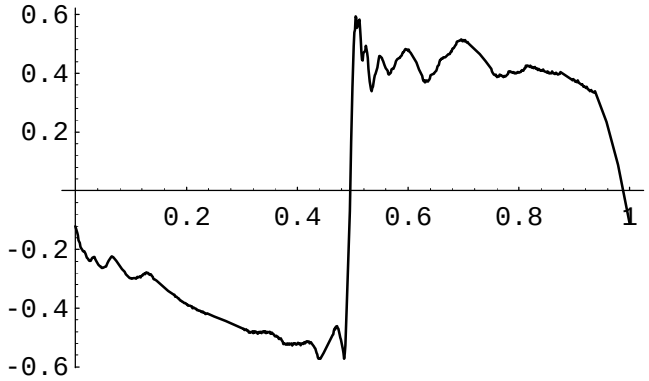
-Graphics-

cc=WT[bb,d4];

dd=ThresSoft[cc, 1.3];

ee=WR[dd,d4];

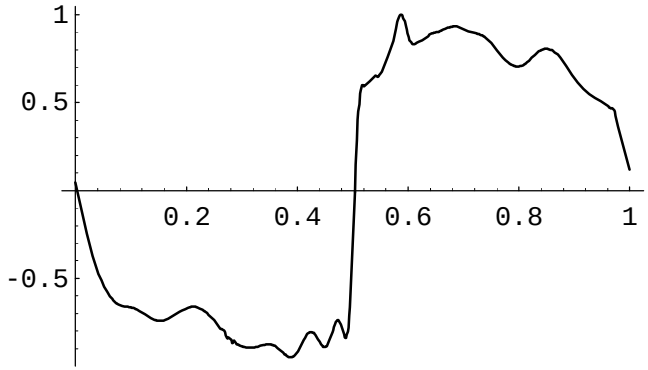
DataPlot[ee]



-Graphics-

ff=WT[bb,d10];

gg=ThresSoft[ff,0.9];
hh=WR[gg, d10];
DataPlot[hh]

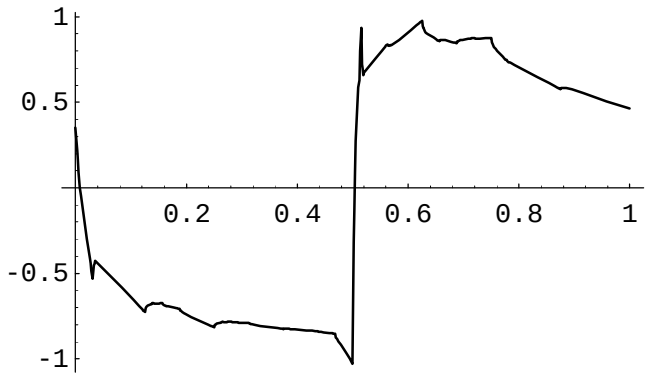


-Graphics-

ii=WT[bb,d2];
jj=ThresSoft[ii,1];

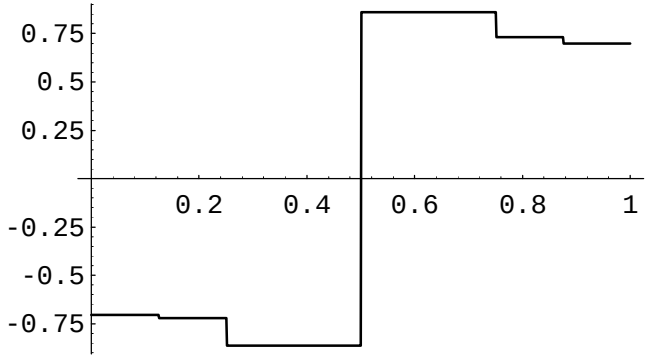
kk=WR[jj, d2]

DataPlot[kk]



-Graphics-

ll=WT[bb,d1];
mm=ThresSoft[ll,0.9];
nn=WR[mm,d1];
DataPlot[nn]



-Graphics-