The à-trous algorithm illustrated

Important remark

Signals and filters are represented as list of (*position,value*)-pairs:

{ { k, a[k] }, { k + 1, a[k + 1] }, { k + 2, a[k + 2] }, ..., { m, a[m] } }

All the signals and filters shown here are finite and can thus be represented using Laurent polynomials.
Filtering operations (i.e. convolutions) are then nothing but multiplications of L-polynomials.

Functions

In[9]:=

```
sig2pol[sig_,z_]:=
(*
Converting signals into L-polynomials
*)
Module[{len},
len=Last[sig][[1]]-First[sig][[1]]+1;
Sum[sig[[k,2]] z^sig[[k,1]],{k,1,len}]
]
```

In[10]:=

```
pol2sig[pol_,z_]:=Module[{low,high},
(*
Converting L-polynomials into signals
*)
low=Exponent[pol,z,Min];
high=Exponent[pol,z,Max];
Table[{k,Coefficient[pol,z,k]
},{k,low,high}]
]
```

In[11]:=

```
convolve[sig_,fil_]:=Module[{sigpol,filpol,outpol,z},
(*
Convolution of the signal sig with the filter fil,
using polynomial multiplication
*)
sigpol=sig2pol[sig,z];
filpol= sig2pol[fil,z];
outpol=sigpol filpol;
pol2sig[outpol,z]
]
```

In[12]:=
```
spread[sig_,m_]:=
(*
Spreading a signal (or filter) by a factor 2^m
which corresponds to m-fold upsampling by a factor 2
*)
Module[{sigpol,outpol,z},
sigpol=sig2pol[sig,z];
outpol=sigpol/.(z→z^(2^m));
pol2sig[outpol,z]
]
```

In[13]:=
```
atrous[sig_,gfil_,hfil_,m_,t_]:=
(*
Applying the à-trous algorithm over m levels to a signal sig
using filters gfil (high-pass filter) and hfil (low-pass filter);
output repeats every row t times (for better visibility);
All the computations are made using polynomial representations
of signals and filters
*)
Module[{left,right,sigpol,gpol,hpol,current,A,out,z},
left=First[sig][[1]];
right=Last[sig][[1]];
sigpol=sig2pol[sig,z];
gpol=sig2pol[gfil,z];
hpol=sig2pol[hfil,z];
current=sigpol*hpol;
A[0]=sigpol*gpol;
out[0]=Table[Coefficient[A[0],z,n],{n,left,right}];
For[k=1,k≤m,k++,
A[k]=  current*((z*gpol)/.(z→z^(2^k)));
out[k]=Table[Coefficient[A[k],z,n],{n,left,right}];
current=current*(hpol/.(z→z^(2^k)))];
Table[out[Quotient[s,t]],{s,0,t(m+1)-1}]
]
```

In[14]:=
```
show[sig_]:=Transpose[sig]//MatrixForm
```

In[15]:=
```
isl=ImageSize→Large;
```

In[16]:=
```
opt[a_]:={Filling→Axis,PlotRange→All,PlotStyle→{PointSize[a],Black}};
```

Signals and polynomials

first signal

In[17]:=
```
signal1 = {{2, a}, {3, b}, {4, c}};
{show[signal1], polynom1 = sig2pol[signal1, z]}
```

Out[17]= $\left\{\begin{pmatrix} 2 & 3 & 4 \\ a & b & c \end{pmatrix}, a z^2 + b z^3 + c z^4\right\}$

second signal

In[18]:=
```
signal2 = {{-1, d}, {0, e}};
{show[signal2], polynom2 = sig2pol[signal2, z]}
```

Out[19]= $\left\{\begin{pmatrix} -1 & 0 \\ d & e \end{pmatrix}, e + \frac{d}{z}\right\}$

convolution of both signals

In[20]:=
```
signal12 = convolve[signal1, signal2]; show[signal12]
```

Out[20]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ a d & b d + a e & c d + b e & c e \end{pmatrix}$$

product of the polynomials

In[21]:=
```
polynom1 * polynom2
```

Out[21]= $\left(e + \frac{d}{z}\right) \left(a z^2 + b z^3 + c z^4\right)$

In[22]:=
```
Collect[polynom1 * polynom2, z]
```

Out[22]= $a d z + (b d + a e) z^2 + (c d + b e) z^3 + c e z^4$

spreading the signals

In[23]:=
```
sp1 = spread[signal1, 2]; show[sp1]
```

Out[23]//MatrixForm=

$$\begin{pmatrix} 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ a & 0 & 0 & 0 & b & 0 & 0 & 0 & c \end{pmatrix}$$

In[24]:=
```
sp2 = spread[signal2, 1]; show[sp2]
```

Out[24]//MatrixForm=

$$\begin{pmatrix} -2 & -1 & 0 \\ d & 0 & e \end{pmatrix}$$

Filtering example

low-pass filter (Haar)

In[25]:=
```
H = {{-1, 1/Sqrt[2]}, {0, 1/Sqrt[2]}}
```

Out[25]= $\left\{\left\{-1, \frac{1}{\sqrt{2}}\right\}, \left\{0, \frac{1}{\sqrt{2}}\right\}\right\}$

high-pass filter (Haar)

In[26]:=
```
G = {{-1, -1/Sqrt[2]}, {0, 1/Sqrt[2]}}
```

Out[26]= $\left\{\left\{-1, -\frac{1}{\sqrt{2}}\right\}, \left\{0, \frac{1}{\sqrt{2}}\right\}\right\}$

signal A

In[27]:=
```
sA = Table[{k, k}, {k, 1, 5}]; show[sA]
```

Out[27]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$$

signal B

In[28]:=
```
sB = Table[{k, -k + 6}, {k, 6, 10}]; show[sB]
```

Out[28]//MatrixForm=

$$\begin{pmatrix} 6 & 7 & 8 & 9 & 10 \\ 0 & -1 & -2 & -3 & -4 \end{pmatrix}$$

signal C

In[29]:=
```
sC = Join[sA, sB]; show[sC]
```

Out[29]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 3 & 4 & 5 & 0 & -1 & -2 & -3 & -4 \end{pmatrix}$$

low-pass filtering the signal C

In[30]:=
```
CH = convolve[sC, H]; show[CH]
```

Out[30]//MatrixForm=

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}+\sqrt{2} & \frac{3}{\sqrt{2}}+\sqrt{2} & \frac{3}{\sqrt{2}}+2\sqrt{2} & \frac{5}{\sqrt{2}}+2\sqrt{2} & \frac{5}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}-\sqrt{2} & -\frac{3}{\sqrt{2}}-\sqrt{2} & -\frac{3}{\sqrt{2}}-
\end{pmatrix}$$

high-pass filtering the signal C

In[31]:=
```
CG = convolve[sC, G]; show[CG]
```

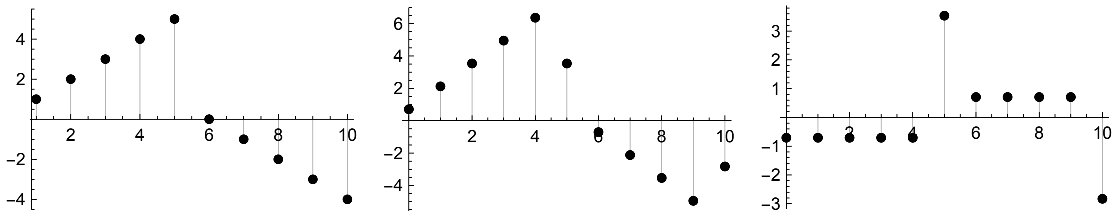
Out[31]//MatrixForm=

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}-\sqrt{2} & -\frac{3}{\sqrt{2}}+\sqrt{2} & \frac{3}{\sqrt{2}}-2\sqrt{2} & -\frac{5}{\sqrt{2}}+2\sqrt{2} & \frac{5}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}+\sqrt{2} & \frac{3}{\sqrt{2}}-\sqrt{2} & -\frac{3}{\sqrt{2}}+
\end{pmatrix}$$

displaying the results

In[32]:=
```
GraphicsRow[{
  ListPlot[sC, opt[0.03]], ListPlot[CH, opt[0.03]], ListPlot[CG, opt[0.03]]}, isl]
```

Out[32]=



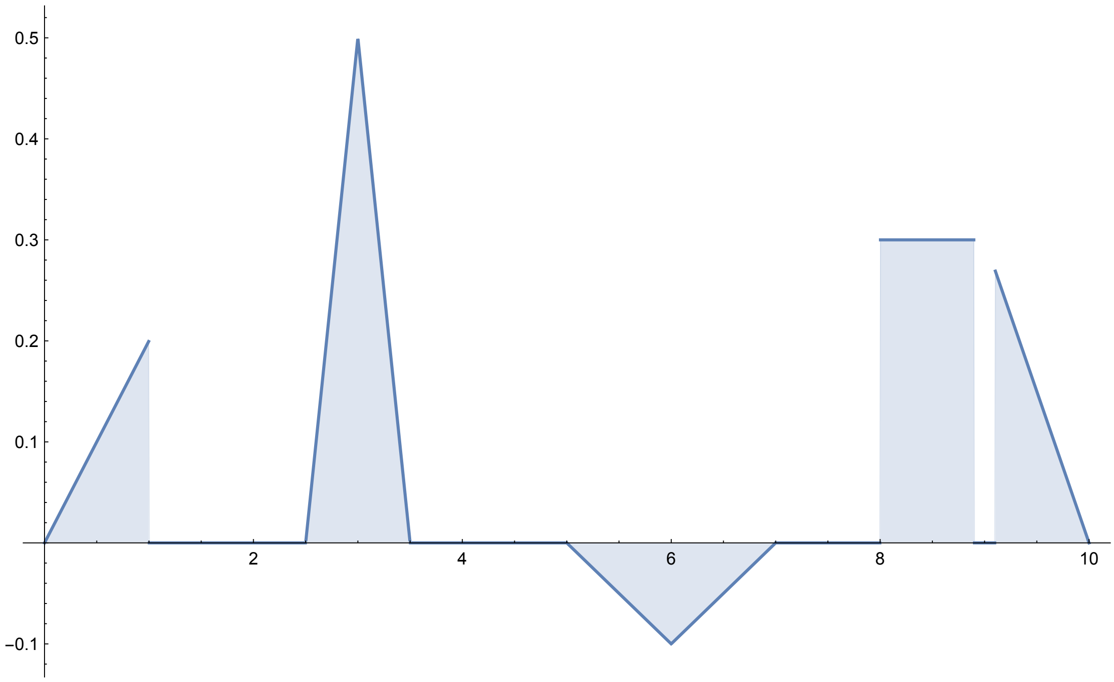
1D à-trous algorithm (example)

defining a piecewise signal

In[33]:=
```
f[x_] := Piecewise[
  {{.2 * x, x < 1},
   {.5 - Abs[3 - x], 2.5 ≤ x && x < 3.5}, {-.1 * (1 - Abs[x - 6]), 5 ≤ x && x < 7},
   {.3, 8 ≤ x && x < 8.9},
   {0.3 (10 - x), 9.1 ≤ x ≤ 10}} ]
```

In[34]:=
```
Plot[f[x], {x, 0, 10}, PlotRange → All, Filling → Axis, ImageSize → Large]
```

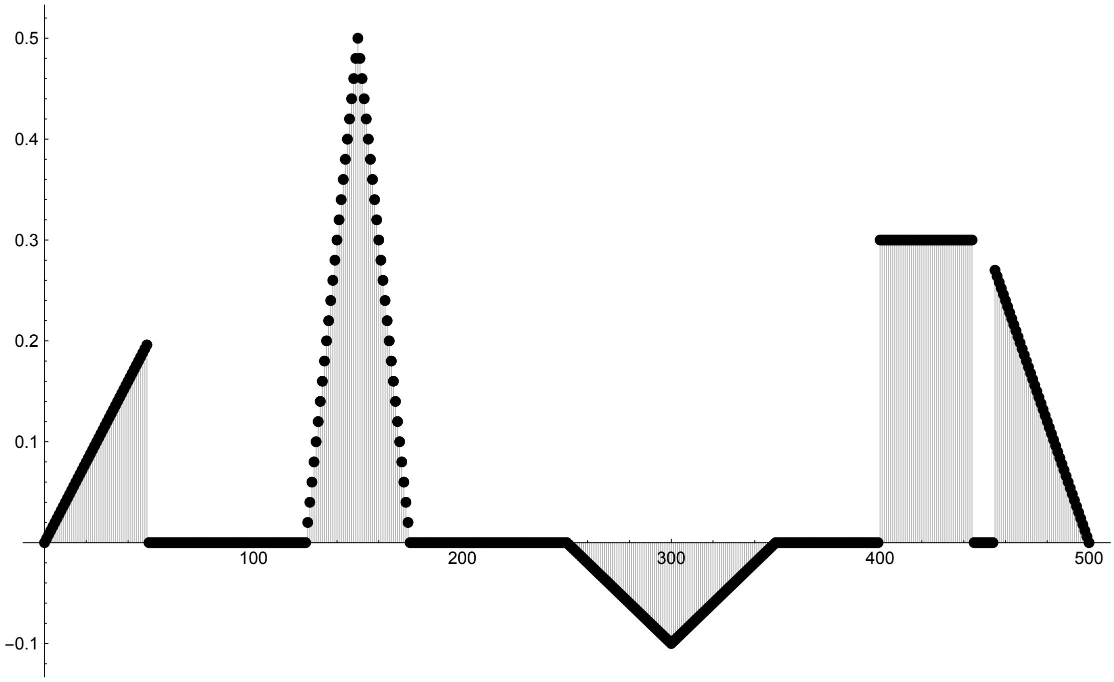
Out[34]=



discretizing the signal

In[35]:= `A = Table[{x, f[x/50]}, {x, 0, 500}]; a = ListPlot[A, opt[0.01], isl]`

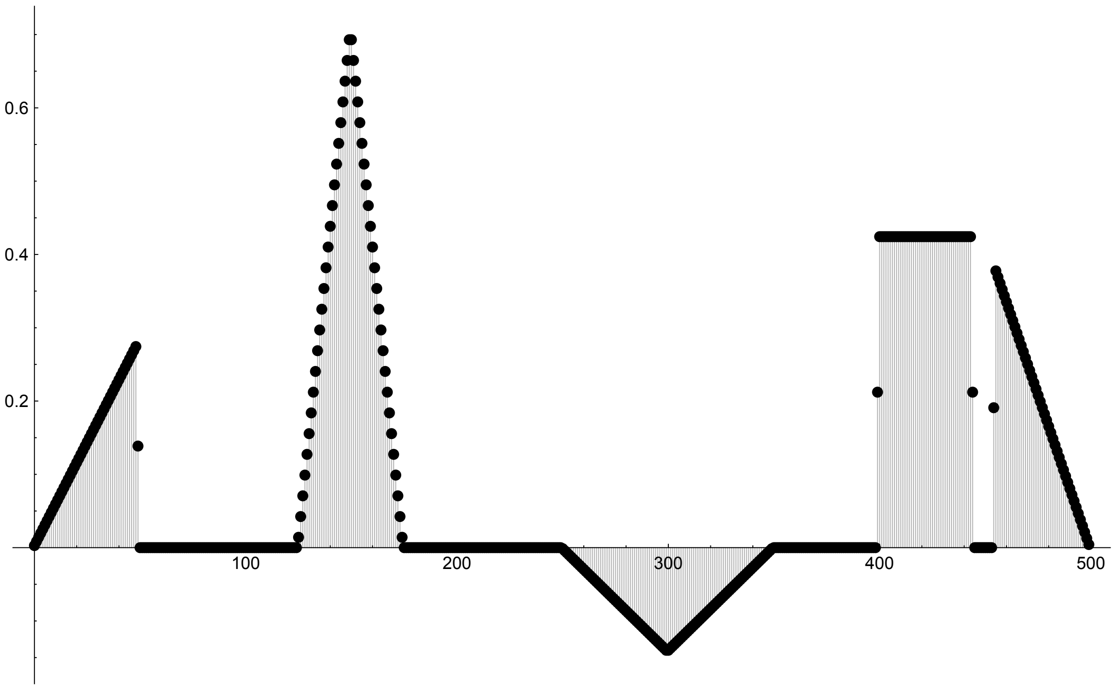
Out[35]=



low-pass filtering of the signal

In[36]:= `AH = convolve[A, H]; ah = ListPlot[AH, opt[0.01], isl]`

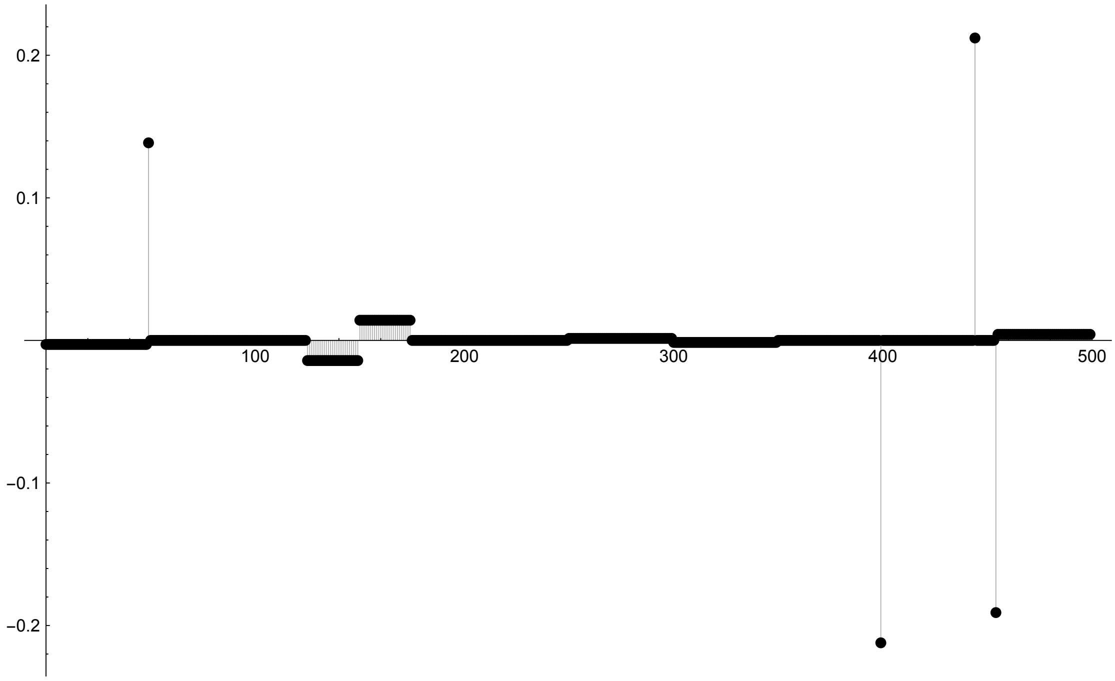
Out[36]=



high-pass filtering of the signal

In[37]:=
```
AG = convolve[A, G]; ag = ListPlot[AG, opt[0.01], isl]
```

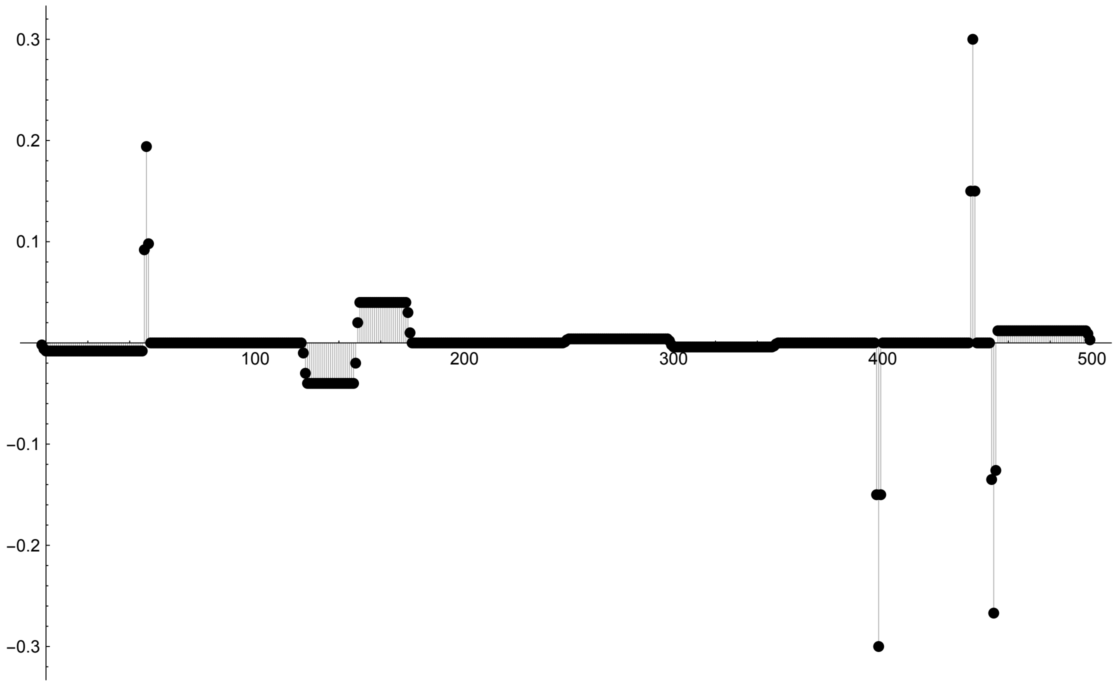
Out[37]=



high-pass filtering the low-pass filtered signal

In[38]:=
```
AHG = convolve[AH, spread[G, 1]]; ahg = ListPlot[AHG, opt[0.01], isl]
```

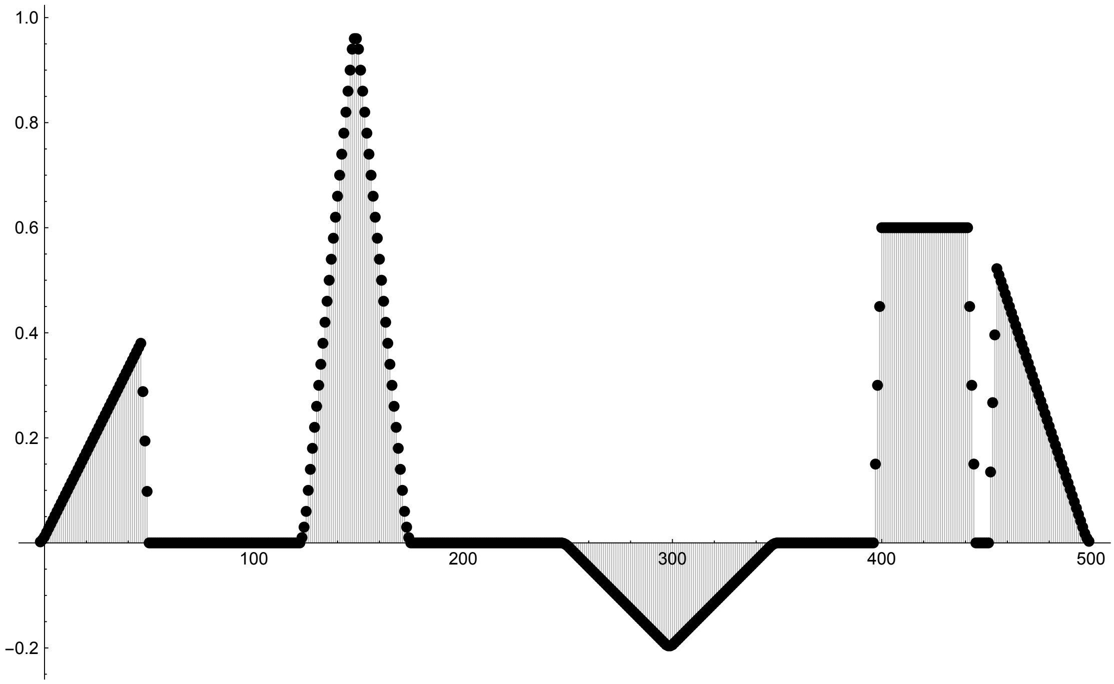
Out[38]=



low-pass filtering the low-pass filtered signal

In[39]:=
```
AHH = convolve[AH, spread[H, 1]]; ahh = ListPlot[AHH, opt[0.01], isl]
```

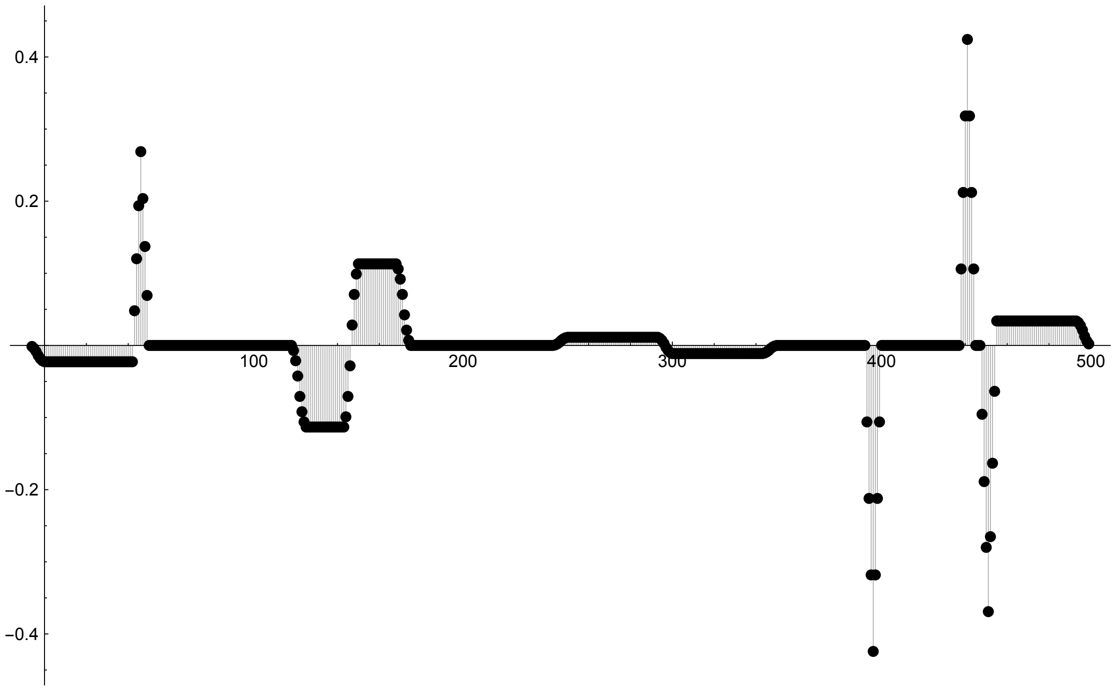
Out[39]=



and more ...

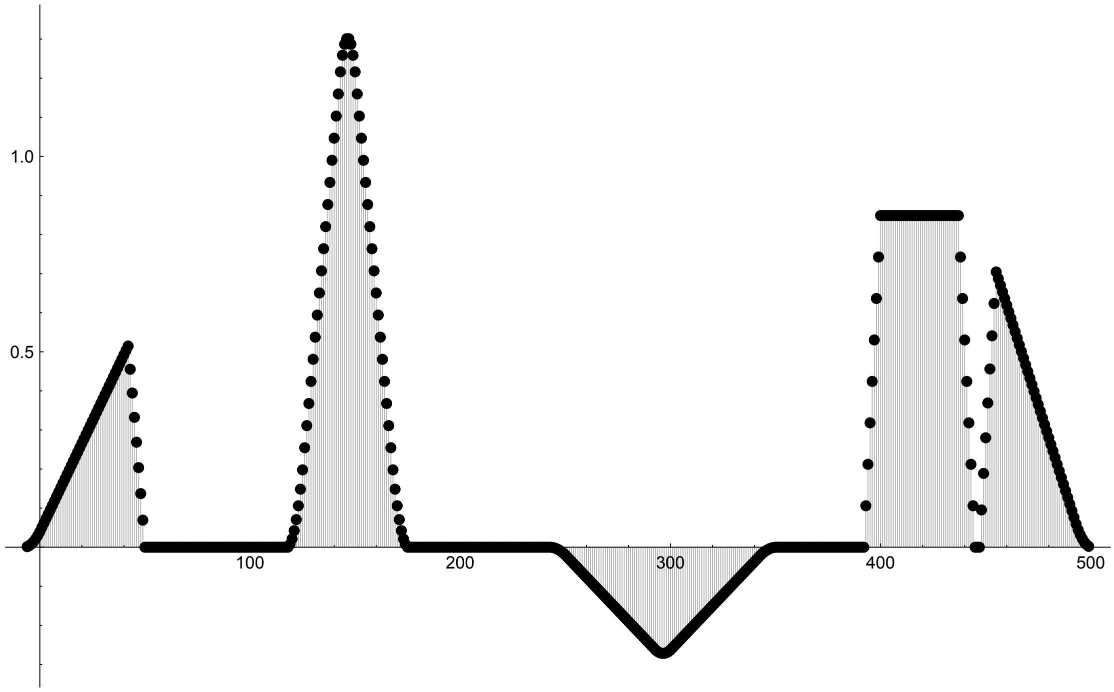
In[40]:=
```
AHHG = convolve[AHH, spread[G, 2]];
ahhg = ListPlot[AHHG, opt[0.01], isl]
```

Out[41]=



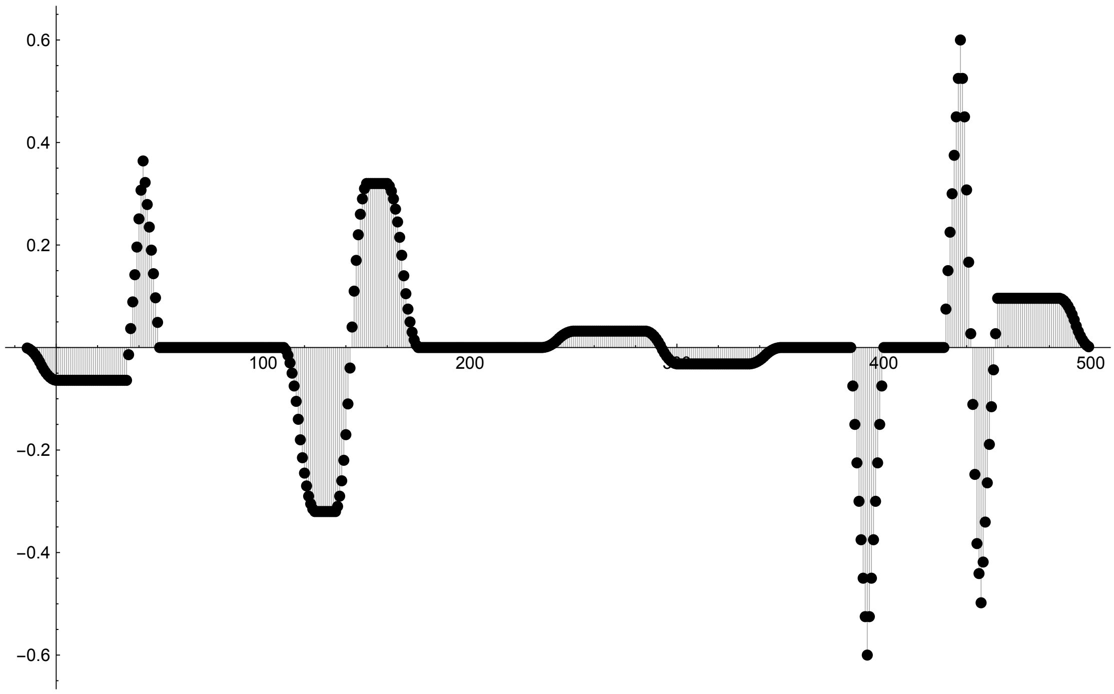
In[42]:=
```
AHHH = convolve[AHH, spread[H, 2]];
ahhh = ListPlot[AHHH, opt[0.01], isl]
```

Out[43]=



In[44]:=
```
AHHHG = convolve[AHHH, spread[G, 3]];
ahhhg = ListPlot[AHHHG, opt[0.01], isl]
```

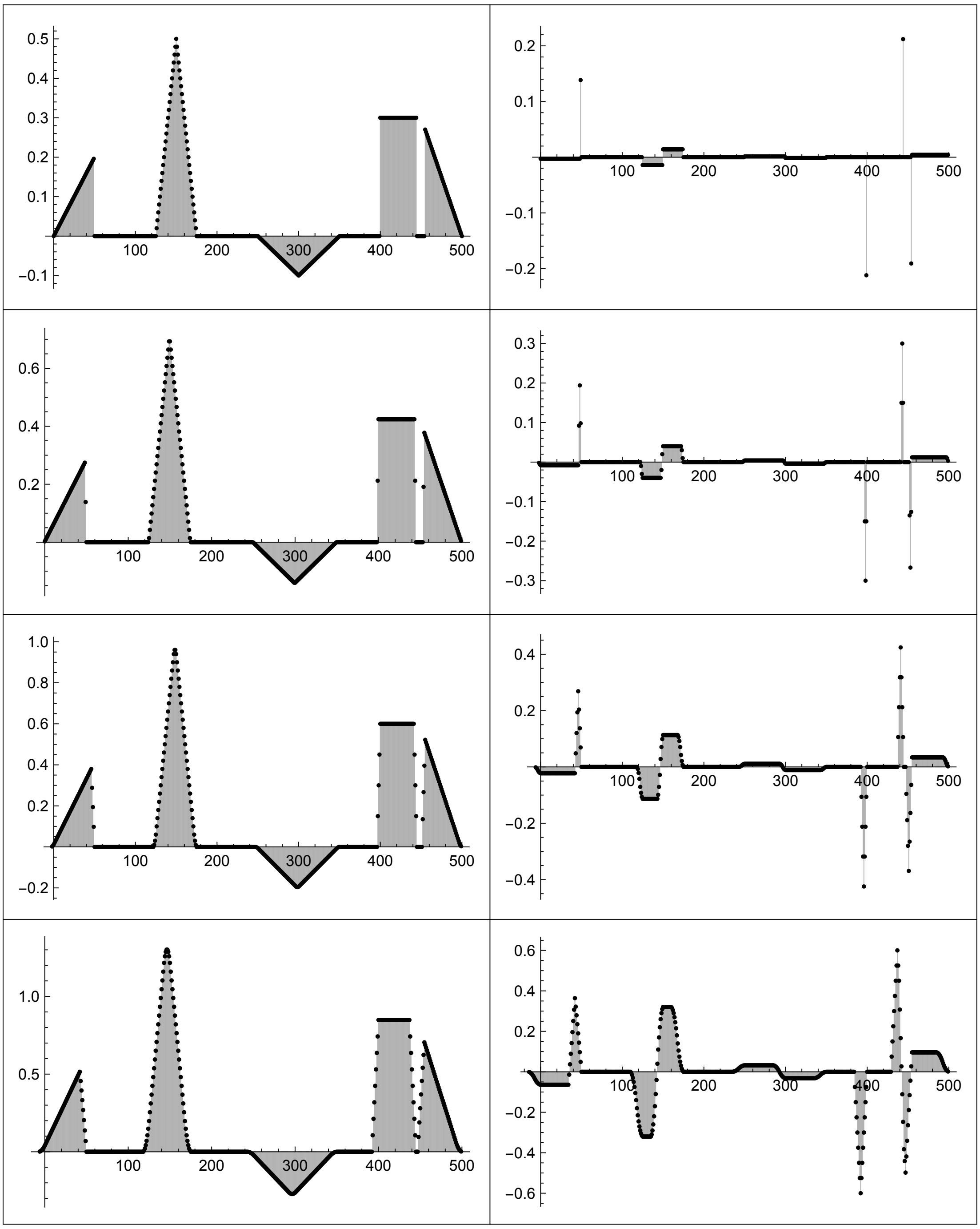
Out[45]=



summarizing the results

In[46]:=
```
GraphicsGrid[{{a, ag}, {ah, ahg}, {ahh, ahhg}, {ahhh, ahhhg}}, Frame → All, isl]
```

Out[46]=



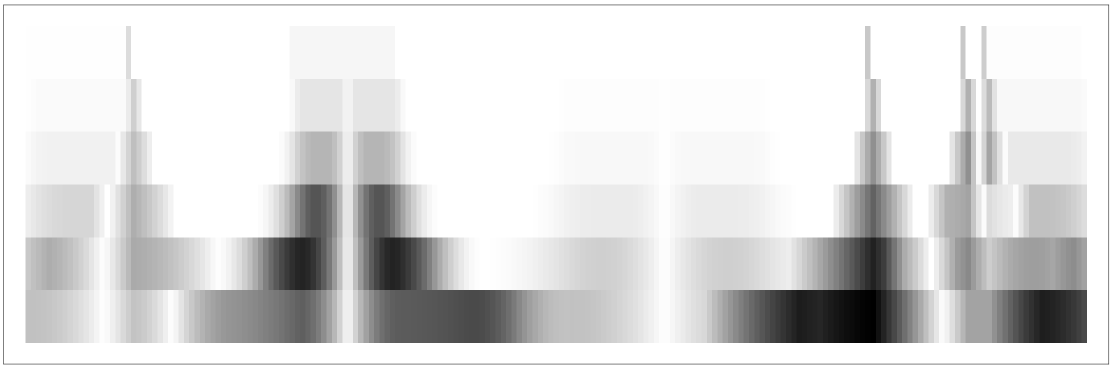
à-trous algorithm (scalogram version)

original signal (discretized)

scalogram after applying a 5 level à-trous algorithm

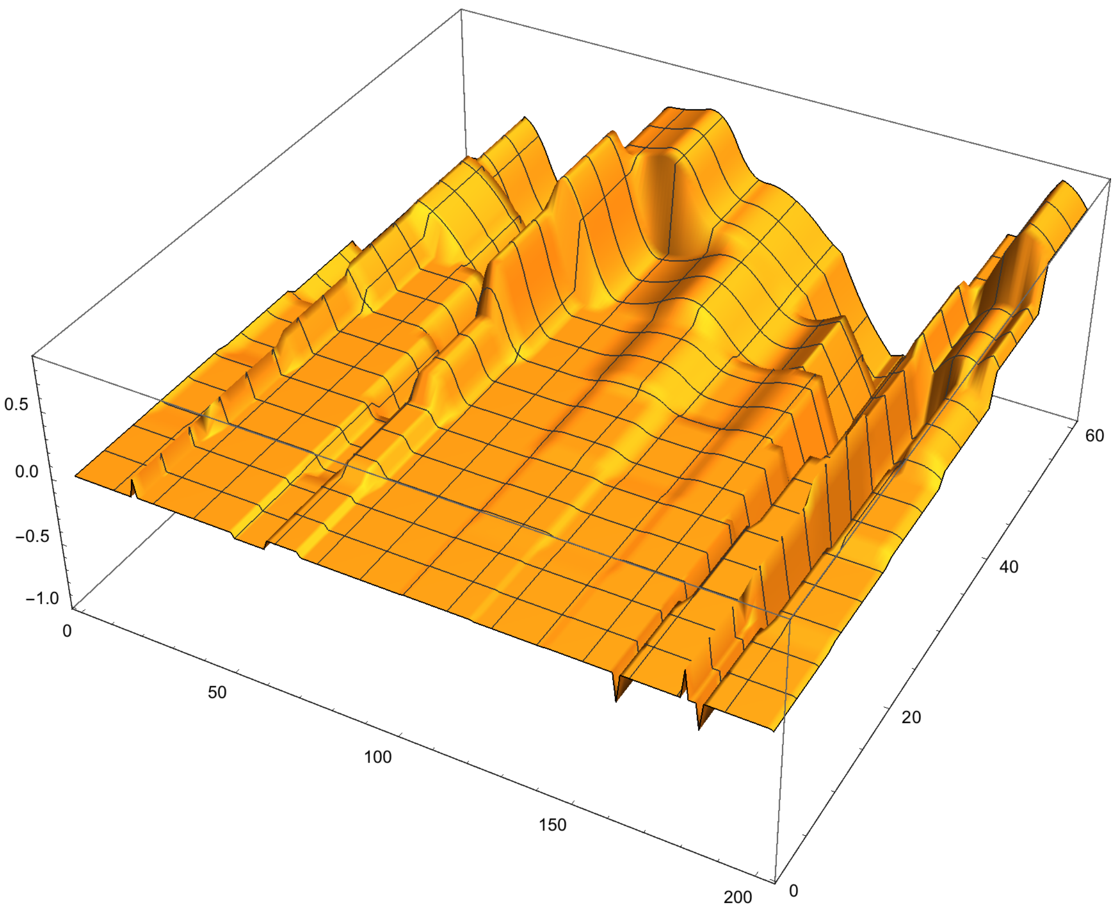
In[47]:=
```
B = Table[{x, f[x/20]}, {x, 0, 200}];
b = atrous[B, G, H, 5, 10];
bplot = ArrayPlot[b, isl]
```

Out[49]=



In[50]:=
```
ListPlot3D[b, PlotRange → All, isl]
```

Out[50]=



noised signal

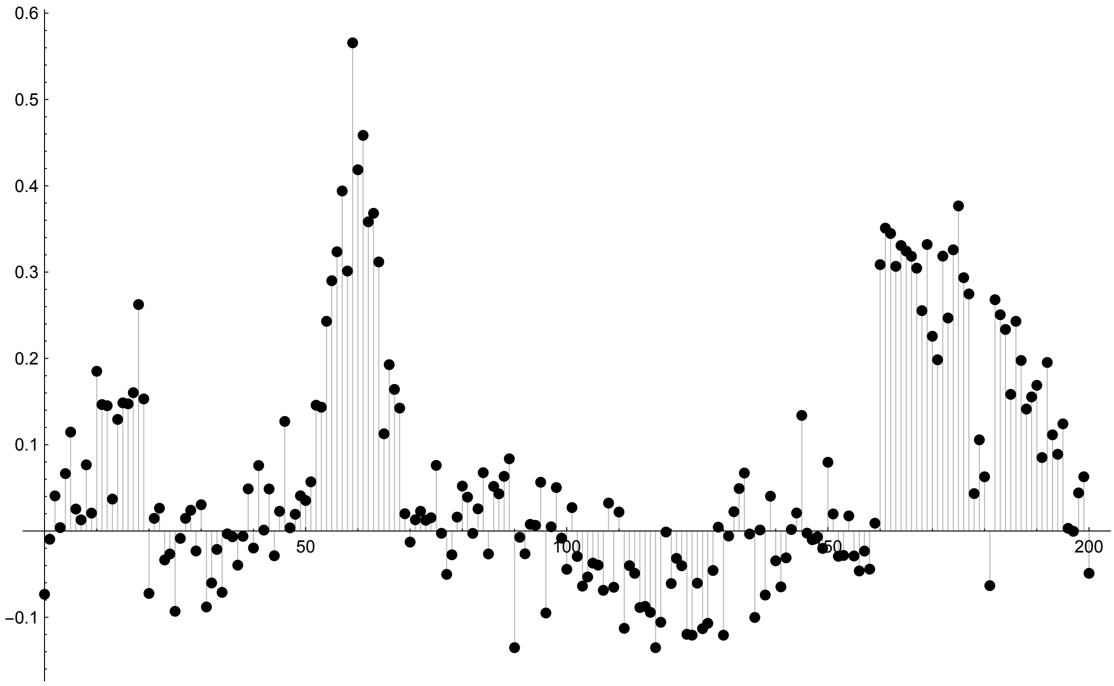
In[51]:=
```
data = RandomVariate[NormalDistribution[0, 0.05], 201];
```

In[52]:=
```
Bn = Table[{x, f[x / 20] + data[[x + 1]]}, {x, 0, 200}];
```

displaying the noised signal

In[53]:=
```
ListPlot[Bn, Filling → Axis, opt[0.01], isl]
```

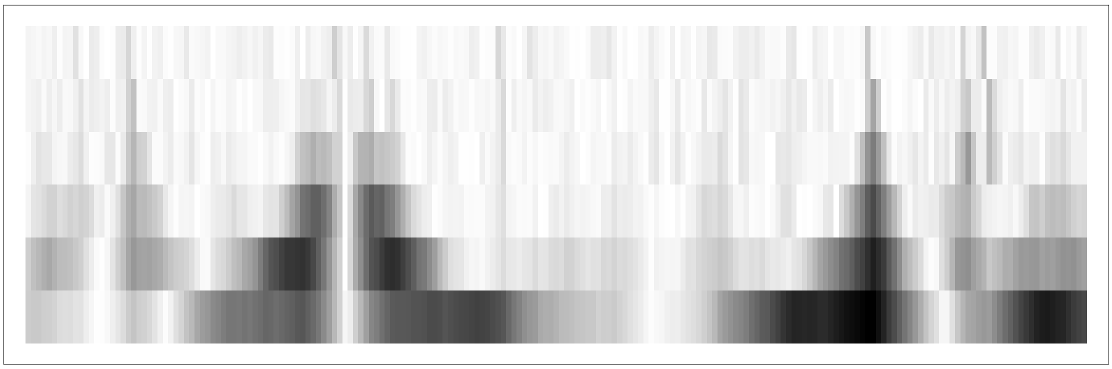
Out[53]=



scalogram after applying a 5 level à-trous algorithm

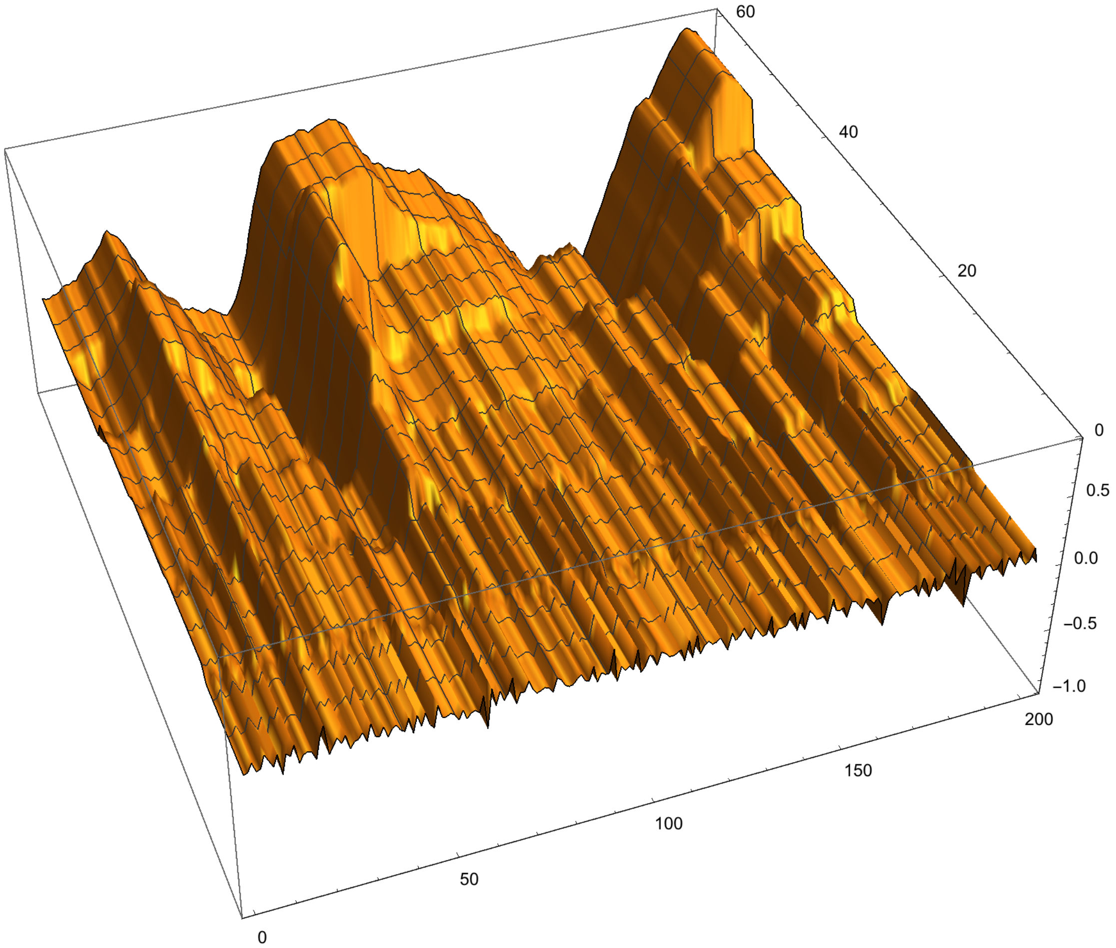
In[54]:=
```
bn = atrous[Bn, G, H, 5, 10];
bnplot = ArrayPlot[bn, isl]
```

Out[55]=



In[56]:= `ListPlot3D[bn, PlotRange → All, isl]`

Out[56]=



applying first smoothing (Bspline filter) to the noised signal

In[57]:= `S[n_] := pol2sig[z^(-n) ((1 + z) / 2)^(2 n), z]`

In[58]:= `S[2]; show[S[2]]`

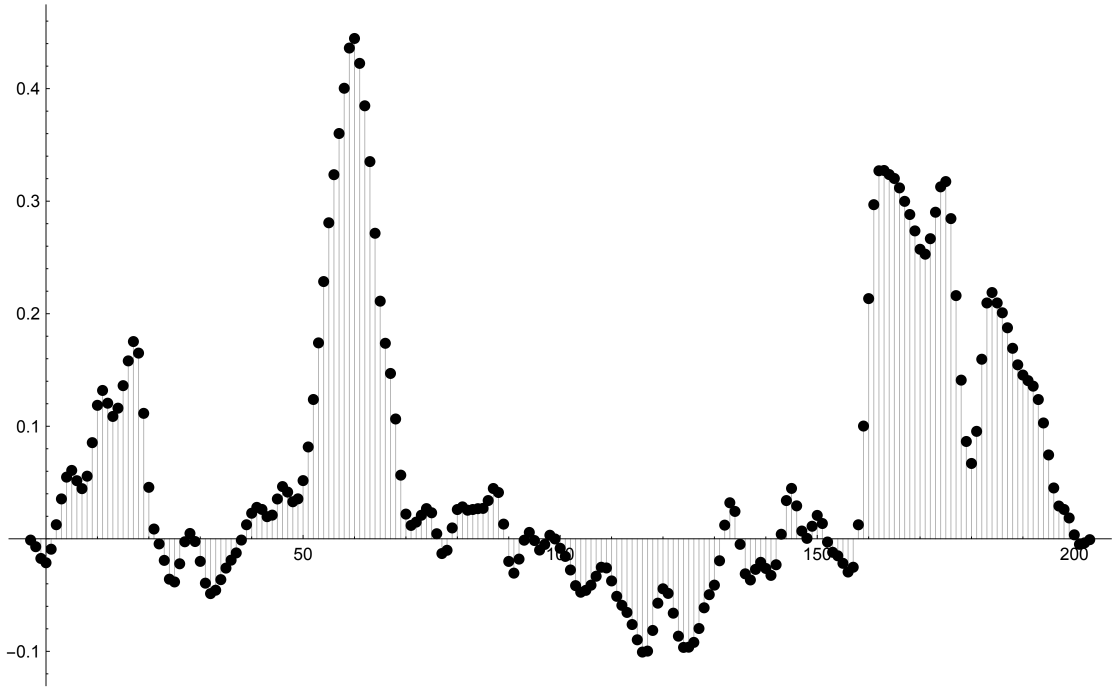
Out[58]//MatrixForm=

$$\begin{pmatrix} -2 & -1 & 0 & 1 & 2 \\ \frac{1}{16} & \frac{1}{4} & \frac{3}{8} & \frac{1}{4} & \frac{1}{16} \end{pmatrix}$$

plotting the smoothed noised signal

In[59]:=
```
Cn = convolve[Bn, S[3]]; ListPlot[Cn, opt[0.01], isl]
```

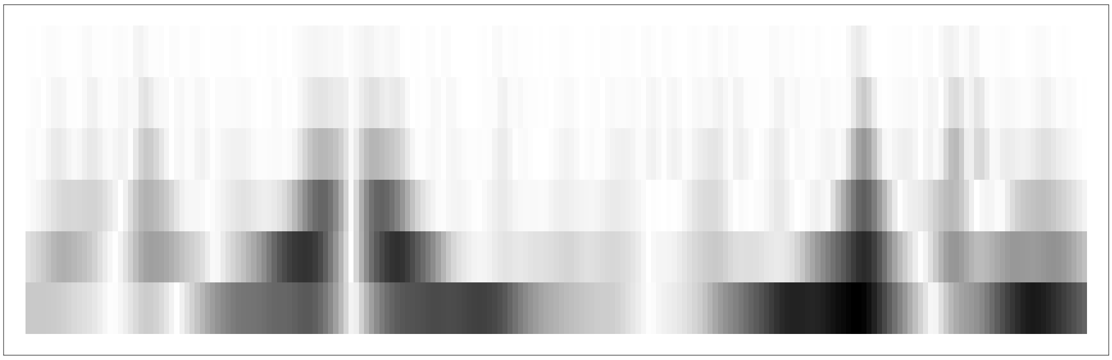
Out[59]=



scalogram after applying a 5 level à-trous algorithm

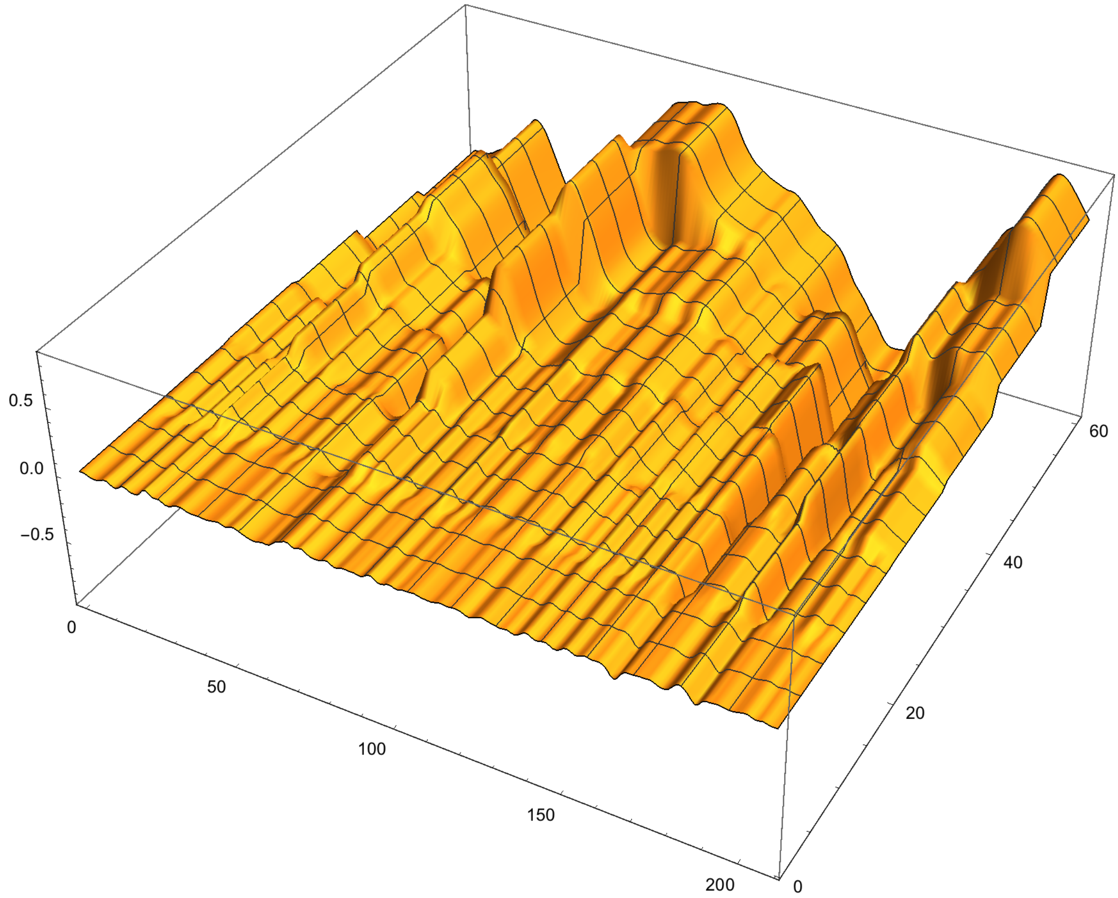
In[60]:=
```
c = atrous[Cn, G, H, 5, 10];
cplot = ArrayPlot[c, isl]
```

Out[61]=



In[62]:= ListPlot3D[c, PlotRange → All, ImageSize → Large]

Out[62]=



comparison of the results

In[63]:=
```
GraphicsGrid[
 {{ListPlot[B, opt[0.01], isl], bplot}, {ListPlot[Bn, opt[0.01] , isl], bnplot},
  {ListPlot[Cn, opt[0.01], isl], cplot}},
 Frame → All, isl]
```

Out[63]=