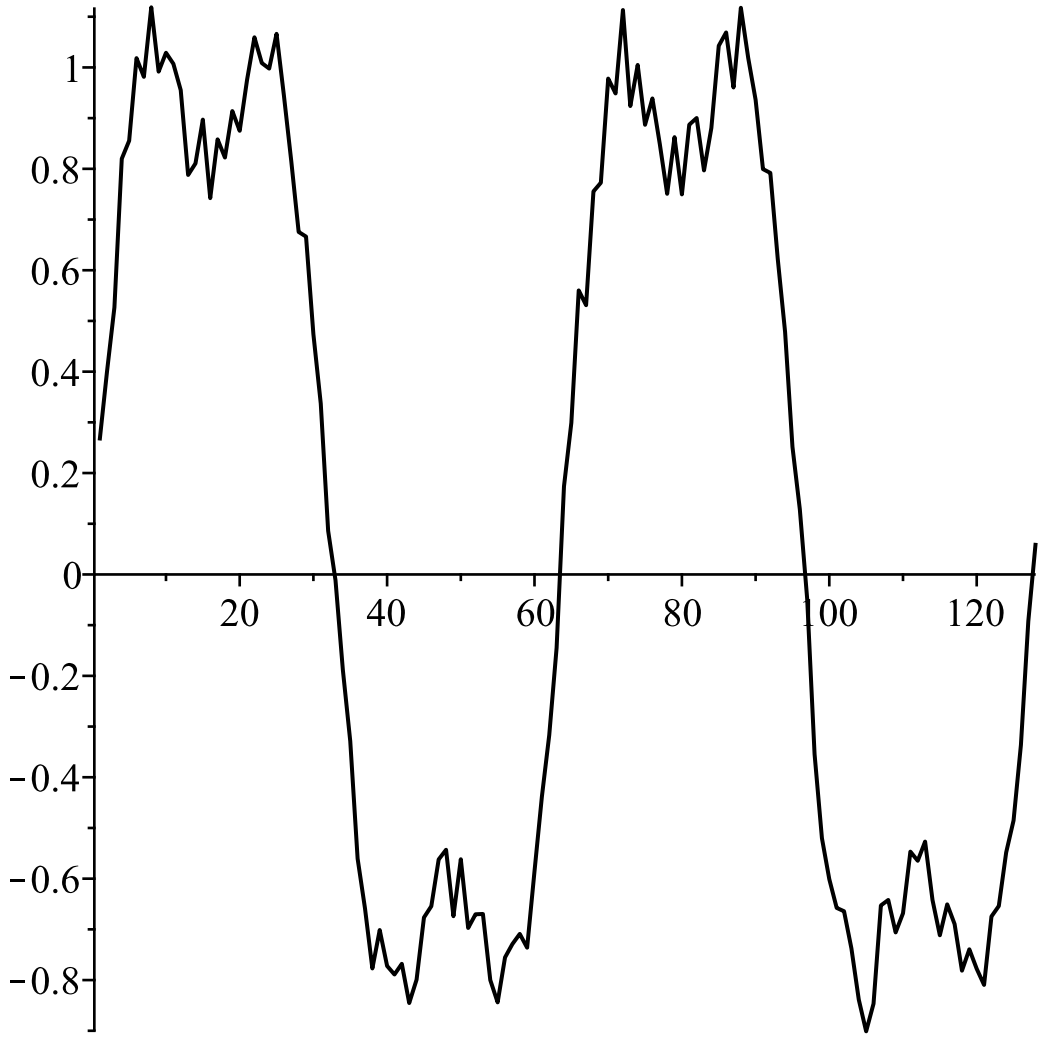
Signal Denoising with Wavelets and FFTs

Load Packages

```
> restart;
  with(SignalProcessing) :
  with(plots) :
```

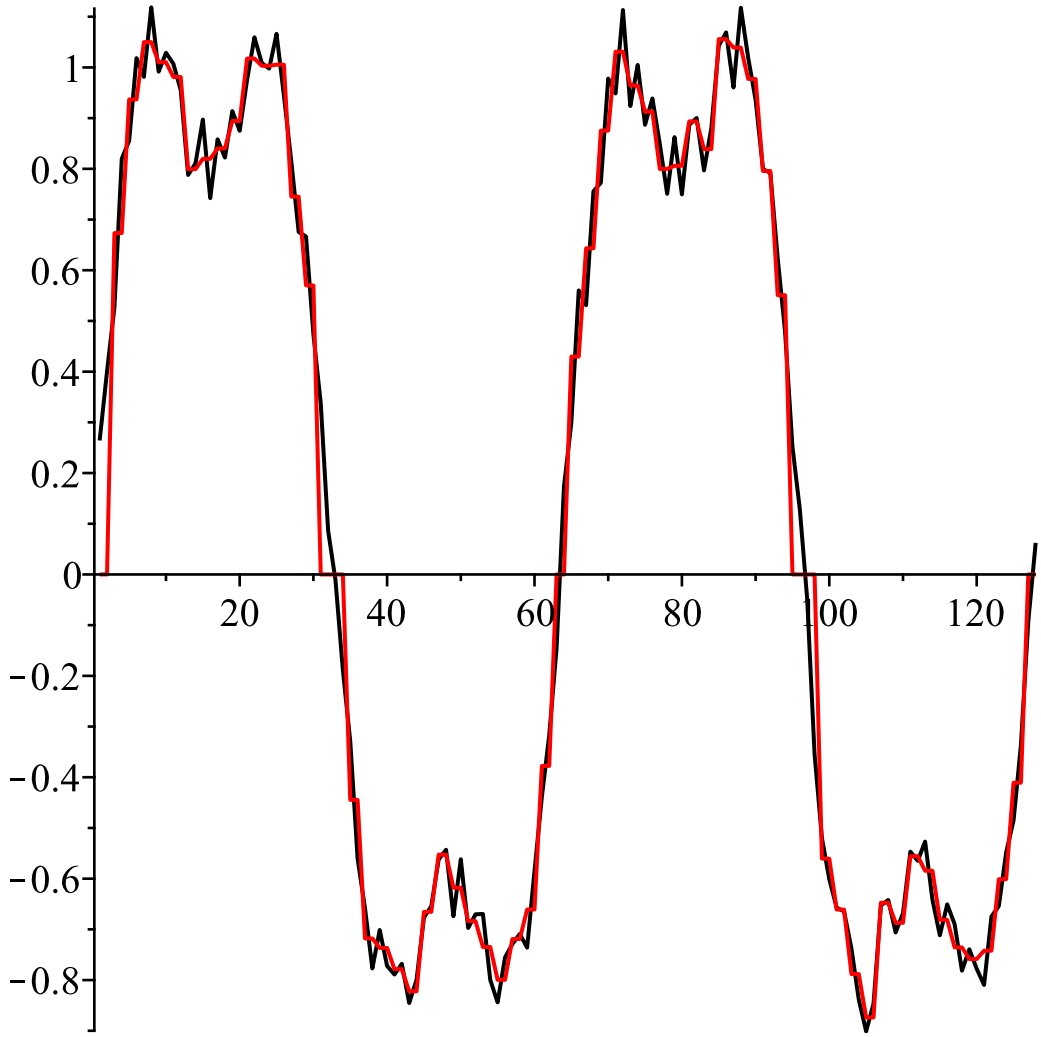
Define Noisy Signal

```
> Signal := Vector(128, i→evalf(sin(4 i π / 128) + 0.3 sin(12 i π / 128)), datatype = float[8]) :
  NS := Vector(128, i→Signal_i + 2·10^(-13) rand( ), datatype = float[8]) :
  p1 := listplot(NS) :
  display(p1)
```



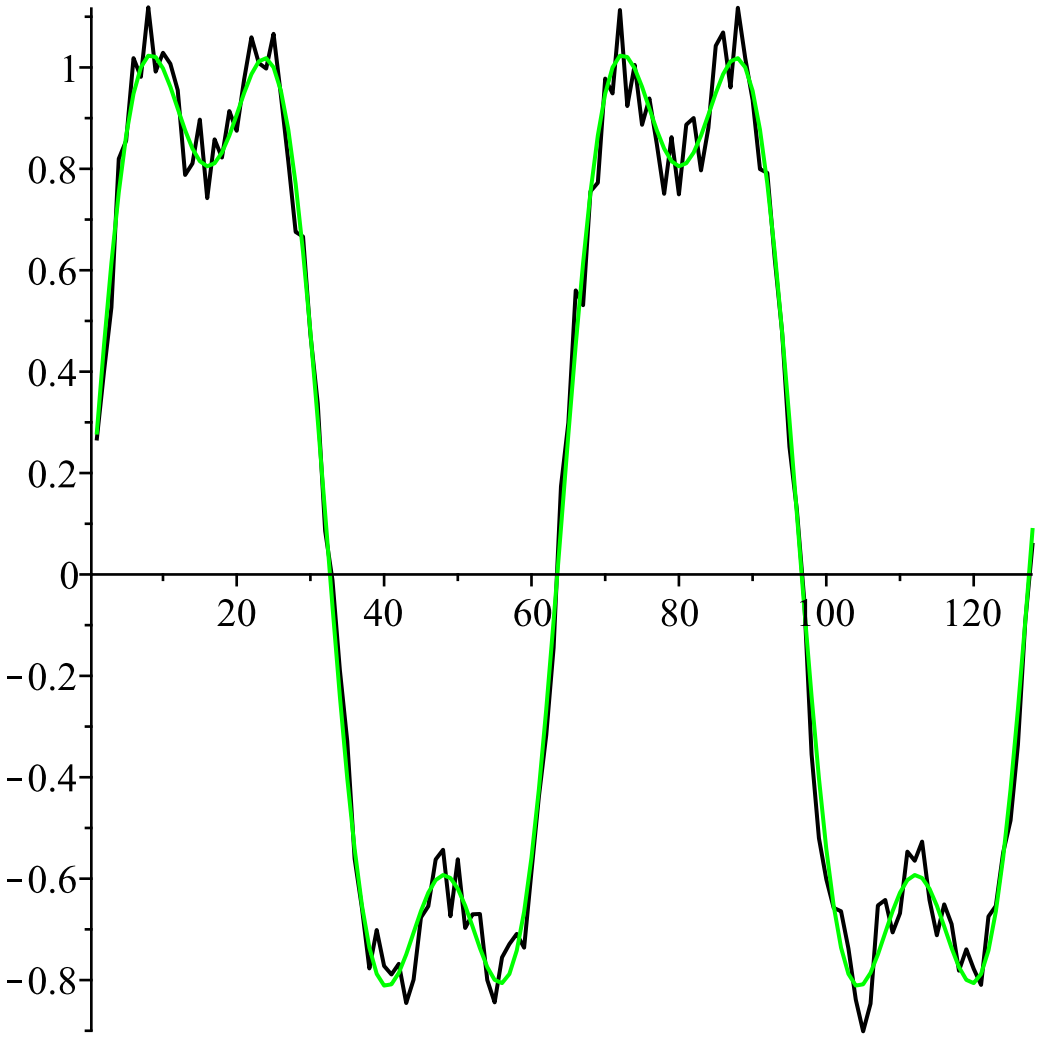
Threshold with Wavelets

```
> thresh := 0.35 :
> c1, c2 := DWT( NS ) :
> c12 := map( z → `if`( |z| ≤ thresh, 0, z ), c1 ) :
> c22 := map( z→`if`( |z| ≤ thresh, 0, z ), c2 ) :
> NS2 := InverseDWT( c12, c22 ) :
> p2 := listplot( NS2, color = red ) :
  display( p1 , p2 )
```



Threshold with FFT

```
> fftData := FFT( NS ) :
> cleanFFTData := map( z → `if`( |z| ≤ thresh, 0, z ), fftData ) :
> cleanData := InverseFFT( cleanFFTData ) :
> p3 := listplot( Re~( cleanData ), color = green ) :
> display( p1, p3 )
```


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
20
40
60
80
100
120

>