Demonstrate noise filtering with Fast Fourier Transform -- with animation.

Instructor: Nam Sun Wang

Generate time series

$N := 2^7 - 1$ $\quad N = 127$ $\quad i := 0 .. N$

$x_i := 0.05 \cdot i$

$f(x) := x \cdot e^{-x}$

$y_i := f(x_i)$

Add noise at a level of $\quad noise := 0.2$

$y\_noise_i := y_i + rnd(noise) - 0.5 \cdot noise$

Notice below how the resulting dimensions from "fft" and "cfft" are different, which, in turn, requires different masks.

Fast Fourier Transform to frequency domain

$Y\_noise := cfft(y\_noise)$

$last(Y\_noise) = 127$

$Y\_noise0 := fft(y\_noise)$

$last(Y\_noise0) = 64$

Make a mask to pass only the low frequencies.

Animation section: toggle off the next equation and set FRAME=0..6 to create a clip.

$FRAME := 2$ $\quad pass := 2^{FRAME}$

$mask_i := (i < pass) + (N - pass < i)$

$Y\_smooth := \overrightarrow{(mask \cdot Y\_noise)}$

$i0 := 0 .. 64$ $\quad mask0_{i0} := (i0 < pass)$

$Y\_smooth0 := \overrightarrow{(mask0 \cdot Y\_noise0)}$

Inverse Fast Fourier Transform back to time domain

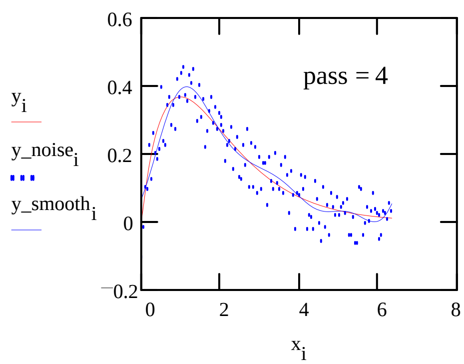
$y\_smooth := icfft(Y\_smooth)$

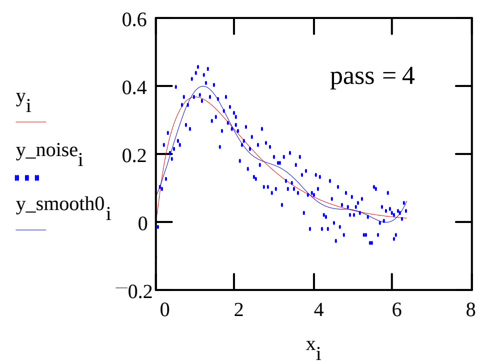
$last(y\_smooth) = 127$

$y\_smooth0 := ifft(Y\_smooth0)$

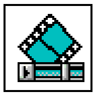
$last(y\_smooth0) = 127$

Plots for comparison





Click on the following icon to play a pre-made animation clip.



smooth.avi

As the number of terms that we allow to pass (pass) decreases, you should see that there is more filtering action and the level of noise decreases; however, both ends of the function become distorted because there is a discontinuity at the boundary as the function is wrapped around.