Subject: Problem discovered in bandpass_filter.pro
Posted by kagoldberg on Wed, 11 Nov 2015 18:58:54 GMT
View Forum Message <> Reply to Message

I just alerted Exelisvis to an error with BANDPASS_FILTER() on IDL 8.4.
I found that for 2D array, the high-frequency cutoff changes by sqrt(2) when the low frequency argument changes from 0 to 0.000001. The program uses different expressions to calculate the filter, based on the lowFreq argument.

Consider the following 2 cases.

**CASE 1
a = randomu(seed, 1000,1000) - 0.5
b = bandpass_filter(a, 0., 0.1, /ideal) ;--- lowFreq is zero
c = abs(fft(b))
window
tvscl, c

**CASE 2
a = randomu(seed, 1000,1000) - 0.5
b = bandpass_filter(a, 0.000001, 0.1, /ideal) ;--- changed lowFreq to something very small
c = abs(fft(b))
window
tvscl, c

Notice that the different lowFreq value here changes the HIGH frequency cutoff
in the output by sqrt(2) because there is an error in the way the function is coded.

In fact, the behavior of the function with lowFrequency NE 0 is incorrect
and leads to cutoff frequencies that are sqrt(2) smaller than they should be.

Say you have a 1000 pixel array, and you set
b = bandpass_filter(a, 0., 0.1, /ideal)

Here, we expect the high frequency cutoff to occur at 0.1 * 1000 = 100 cycles.
Instead, a quick test will show that the cutoff occurs at 70 cycles ~ 100/sqrt(2).
This occurs with /butterworth and /ideal, maybe /gaussian but I didn't test it.

I discovered it in the difference that occurs with filtered an array using a BUTTERWORTH() and 2 FFTs, versus just using BANDPASS_FILTER(... BUTTERWORTH=N)