
Subject: Re: bug in IDL's hanning() window-generating function

Posted by [david\[2\]](#) on Wed, 01 Aug 2001 12:20:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

Scott Bennett writes after a long analysis of the Hanning function:

> In any case, I think the point Harris made is that a discrete
> sampling of a window function should not taper to the same value at
> the end that it has at the beginning because to do so would include
> the first sample of the *next* period (windowed segment.) So IDL's
> hanning() gets it wrong for both even- and odd-length windows. :-(

Uh, huh. And how did RSI respond when you contacted them
about it?

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438 E-Mail: davidf@dfanning.com

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: bug in IDL's hanning() window-generating function

Posted by [Jaco van Gorkom](#) on Wed, 01 Aug 2001 14:30:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Scott Bennett writes after a long analysis of the Hanning function:

>

>> In any case, I think the point Harris made is that a discrete
>> sampling of a window function should not taper to the same value at
>> the end that it has at the beginning because to do so would include
>> the first sample of the *next* period (windowed segment.) So IDL's
>> hanning() gets it wrong for both even- and odd-length windows. :-(

>

> Uh, huh. And how did RSI respond when you contacted them
> about it?

I suspect they would suggest the following workaround:

```
function Harris, n, _EXTRA=extra
  return, (hanning(n+1, _EXTRA=extra))[0:n-1]
end
```

Scott, I've read your post, but I'm not sure I'm getting the point. Tapering from zero to zero seems like good idea to me, the symmetry sort of "feels good". Besides, it is convenient to have the weight of the window at its centre. What exactly is so wrong with it?

Jaco

PS: I would look it up myself, but I seem to have misplaced IEEE vol. 66 somehow...

Jaco van Gorkom
FOM-Instituut voor Plasmafysica "Rijnhuizen", The Netherlands
e-mail: gorkom@rijnh.nl

Subject: Re: bug in IDL's hanning() window-generating function
Posted by [Paul van Delst](#) on Wed, 01 Aug 2001 15:10:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

Jaco van Gorkom wrote:

```
>
> David Fanning wrote:
>> Scott Bennett writes after a long analysis of the Hanning function:
>>
>>> In any case, I think the point Harris made is that a discrete
>>> sampling of a window function should not taper to the same value at
>>> the end that it has at the beginning because to do so would include
>>> the first sample of the *next* period (windowed segment.) So IDL's
>>> hanning() gets it wrong for both even- and odd-length windows. :-(
>>
>> Uh, huh. And how did RSI respond when you contacted them
>> about it?
>
> I suspect they would suggest the following workaround:
>
> function Harris, n, _EXTRA=extra
>   return, (hanning(n+1, _EXTRA=extra))[0:n-1]
> end
>
> Scott, I've read your post, but I'm not sure I'm getting the point.
> Tapering from zero to zero seems like good idea to me, the symmetry
> sort of "feels good". Besides, it is convenient to have the weight of
> the window at its centre. What exactly is so wrong with it?
```

Hello,

I think Scott's post underscores the fact that any sort of Fourier analysis is not to be

undertaken lightly without fully understanding how any code used works (or is supposed to work) and knowing, in some fashion, what the result should look like in advance. The difference of a point (or a window being a little bit wider etc) can make a *huge* difference in how one interprets or subsequently uses the results in FFT applications. While I'm not dismissing your comments about the good vibrations one typically gets from symmetry (I try to achieve that as much as possible too) I'm going to file Scott's message under the "READ_THESE_FIRST!!!" sub-directory in my FFT code/work directory. :o)

paulv

--

Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
 Alexander Pope.

Subject: Re: bug in IDL's hanning() window-generating function
Posted by [Brian Jackel](#) on Wed, 01 Aug 2001 17:51:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Scott

A friend and I noticed this several years ago. My recollection is that he reported this to RSI, but they weren't convinced of the problem. Of course, it *is* a (minor) problem.

At the time I was playing around with windows (or apodizing functions, or tapers) and wrote some IDL code. It's included below. The "Slepian" window is commented out because it requires something clever with eigenvalues, and the details have escaped me.

To see what Scott is talking about, try typing the following two commands:

```
test= FFT_TAPER(512,'hanning',/PLOT) ;correct way
test= FFT_TAPER(512,'badhanning',/PLOT) ;IDL way
```

Although this is old code, I would appreciate any comments or bug reports. Have fun.

Brian Jackel

```
;=====
;distributed freely,
```

```

;use at own risk,
;send all bug reports to bjackel@phys.ucalgary.ca
;+
; NAME: FFT_TAPER
;
; PURPOSE: This function provides a selection of "tapers" (also known
;          as "windows" or "apodizing functions") to reduce sidelobe
;          leakage when calculating FFT's.
;
; CATEGORY: Time Series Analysis, Signal Processing
;
; CALLING SEQUENCE:  Result=
FFT_TAPER(Nelements,Windowname,[Parameter])
;
; INPUTS:
;   Nelements  the number of elements in the window ie. # of
;               elements in the time series to be windowed.
;               Must be a positive integer (floats will be rounded).
;
;   Windowname  a string containing the name of the windowing function
;               to use. Not case sensitive, whitespace will be
;               ignored.
;
; OPTIONAL INPUTS:
;   Parameter  some tapers can be tuned with a parameter value. See
;               specific descriptions for details.
;
; KEYWORD PARAMETERS:
;
;   PLOT_EXAMPLE  if set, the window and amplitude spectrum will
;                 be plotted
;   RETURN_WINDOWLIST  if set, Result will be a string array of all
;                 valid Windownames
;
;
;
;
;
;
; Kaiser-Bessel  This taper can be tuned. Set the parameter to
; (Nuttall p89)  the desired location (in scaled Lf/pi units) of
;                 the first null. ie. 2, 3, or 4
;
;
; van der Maas  This taper can be tuned. Set the parameter to
; (Nuttall p90)  the ratio of the log of the peak to sidelobe level
;                 ie. 2 (factor of 100 or 20dB) or 3 (30dB)
;
;
;
;

```

; MODIFICATION HISTORY:

; Written by: Brian Jackel 1995

; -

Function FFT_TAPER,Length,Windowtype,Parameter, \$
 RETURN_WINDOWLIST=return_windowlist,PLOT_EXAMPLE=plot_examp l e

;

; Make a list of all the legal window names, and put them in an array.

If the

; keyword RETURN_WINDOWLIST is set, then just return the array of
strings. This

; is useful for letting the user know what options are available.

;

 windowlist=['Hanning', 'Hamming', 'Blackman', 'Exact Blackman',
'Blackman-Harris']

 windowlist= [windowlist, 'Dolph-Chebyshev', 'Kaiser-Bessel', 'van der
Maas', 'Slepian']

 IF KEYWORD_SET(RETURN_WINDOWLIST) THEN return,windowlist

; First, test Length:

; If not set, use 16.

; If it's a scalar, that's the window length.

; If it's a 1D array, then use the # of elements as the window length.

; Otherwise, return with an error message.

;

IF (N_PARAMS() LT 1) THEN length= 16

siz= SIZE(length)

CASE siz(0) OF

0: L= length

1: L= N_ELEMENTS(length)

ELSE: MESSAGE,'Length must be a scalar or a 1D array'

ENDCASE

; Check the Windowtype.

; If none given, use Hanning

; Remove spaces, convert to uppercase

;

IF (N_PARAMS() LT 2) THEN BEGIN

 MESSAGE,'No Windowtype given, using default: Hanning',/INFORMATIONAL

 Windowtype= 'Hanning'

ENDIF

Windowtype= STRUPCASE(STRCOMPRESS(Windowtype,/REMOVE_ALL))

print,windowtype

; If no additional parameter is provided, set it to 1

```

;
IF (N_PARAMS() LT 3) THEN parameter=1.0

result= 1.0
nn= 2.0*!pi*FINDGEN(L)/L
t= 2.0*FINDGEN(L)/L - 1.0      ;-1 to almost 1

CASE (windowtype) OF

;-----
;The Hanning window is widely used, due to the advantages of a simple
; representation in the time domain and an analytic expression for the
; transform. Unfortunately, it has neither a particularly narrow main
; lobe nor low sidelobes.
;
'HANNING': wind= 1.0 + cos(!pi*t)

;-----
;Although the analytic expression of a taper is (usually) symmetric,
the
; FFT form shouldn't be, as the last point is the same as the first,
and
; shouldn't be explicitly included. Here's what the Hanning window
looks
; like if done incorrectly. Not a big difference, but worth getting
right.
;
'BADHANNING': BEGIN
    t= 2.0*FINDGEN(L)/(L-1.0) - 1.0
    wind= 1.0 + cos(!pi*t)
END

;-----
;The Hamming window has a time domain representation similar to the
Hanning.
;It has a comparable mainlobe width, but a lower first sidelobe.
;
;
'HAMMING': wind= 1.0 + 0.46/0.54*cos(!pi*t)

;-----
;The Blackmann window has a moderately wide main lobe, and low
sidelobes
;

```

```
'BLACKMAN':wind= 1.0 + 0.50/0.42*cos(!pi*t) + 0.08/0.42*cos(2.0*!pi*t)
```

```
-----  
;  
;  
;  
;
```

```
"EXACTBLACKMAN": wind= 1.0 + 1.16402*cos(!pi*t) +  
0.180146*cos(2.0*!pi*t)
```

```
'DOLPH-CHEBYSHEV':BEGIN
```

```
    order= L  
    x= 10.0^parameter  
    z0 = COSH( ALOG(x + SQRT(x^2-1.0) ) / order )
```

```
;Harris calls this 'beta'
```

```
    arg= z0 * COS(nn/2.0)  
    temp= (-1)^FINDGEN(L) * TCHEBYSHEV(arg,order) / x  
    ; stop  
    wind= FLOAT(FFT(temp,1))  
END
```

```
'KAISER-BESSEL':BEGIN
```

```
    b= parameter*!pi  
    arg= SQRT( (1.0-t^2) > 0.0)          ;avoid
```

```
problems at arg^2 = 1
```

```
    wind= BESELI(b*arg,0) / (SINH(b)/b)  
    ; stop  
END
```

```
'NORTON-BEER':BEGIN
```

```
    n= FIX(parameter)  
    CASE n OF                      ;four possible
```

```
window types (0,1,2,3)
```

```
    0: c= [1.0, 0.0, 0.0, 0.0, 0.0]  
    1: c= [0.384093, -0.087577, 0.703484, 0.0, 0.0]  
    2: c= [0.152442, -0.136176, 0.983734, 0.0, 0.0]  
    3: c= [0.045335, 0.0, 0.554883, 0.0, 0.399782 ]
```

```
    ELSE:MESSAGE,'For Norton-Beer, parameter should be an  
integer from 0 (no apodization) to 3 (strong apodization)'
```

```
    ENDCASE  
    u= 2.0 * FINDGEN(L) / L - 1.0  
    arg= 1.0 - u^2  
    wind= c(0)  
    FOR i=1,4 DO BEGIN  
        IF ( c(i) NE 0.0 ) THEN wind= wind + c(i)*arg^i  
    ENDFOR  
END
```



```

IF KEYWORD_SET(PLOT_EXAMPLE) THEN BEGIN
  !p.multi=[0,1,2]
  spec= 4.0*ABS(FFT([result,REPLICATE(0.0,l*3)],-1))
  spec= 10.0*ALOG10(spec)
  xspec= 2.0*FINDGEN(l*4.0)/(4.0*l)
  every4= FINDGEN(l)*4
  PLOT,t,result,TITLE=windowtype + ' Window',PSYM=-4,SYMSIZE=0.5
  XYOUTS,0.7,!y.crange(1)*0.85,STRING(l,FORMAT='("N=",l3)')
  IF (parameter(0) NE 1) THEN BEGIN
    pstring= STRCOMPRESS( STRING(parameter(0),FORMAT='("p=",F6.1)'))
    XYOUTS,0.7,!y.crange(1)*0.75,pstring
  ENDIF
  !p.multi=[2,2,2]
  PLOT,xspec,spec,TITLE='Amplitude
Spectrum',YRANGE=[-100,0],XRANGE=[0,0.5],XSTYLE=1,YTITLE='dB ',XTITLE='f
/fN',XTICKS=2
  PLOT,xspec*l,spec,TITLE='Amplitude
Spectrum',YRANGE=[-50,0],PSYM=-4,SYMSIZE=0.3,XRANGE=[0,12],X
STYLE=1,XTICKS=2,XTITLE='Delta
f'
  OPLOT,xspec(every4)*l,spec(every4),PSYM=4,SYMSIZE=0.7
ENDIF

RETURN,result
END

```

Subject: Re: bug in IDL's hanning() window-generating function
 Posted by [bennetsc](#) on Thu, 02 Aug 2001 05:56:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <MPG.15d1a1e0a012bdcd989e44@news.frii.com>,
 David Fanning <david@dfanning.com> wrote:
 > Scott Bennett writes after a long analysis of the Hanning function:

```

>
>>   In any case, I think the point Harris made is that a discrete
>> sampling of a window function should not taper to the same value at
>> the end that it has at the beginning because to do so would include
>> the first sample of the *next* period (windowed segment.) So IDL's
>> hanning() gets it wrong for both even- and odd-length windows. :-(

```

```

>
> Uh, huh. And how did RSI respond when you contacted them
> about it?
>

```

I didn't contact them. I'm disgusted with them after seeing how they cheated my faculty advisor and, when I asked them to rectify

their errors, they refused to do so. I recommended to my advisor that he turn the whole matter over to the university's legal counsel. He told me that he was not going to renew the service contract any more, so I doubt he will bother with having the legal counsel deal with it, but either way it will be his decision and it's now out of my hands. Given their denial of service while we had a service contract, I doubt they would do any better now that we no longer have a service contract.

Sciences
Scott Bennett, Comm. ASMELG, CFIAG
College of Oceanic and Atmospheric
Oregon State University
Corvallis, Oregon 97331

```
*****
* Internet:      sbennett at oce.orst.edu          *
* -----*
* "Lay then the axe to the root, and teach governments humanity.  *
* It is their sanguinary punishments which corrupt mankind."      *
* -- _The_Rights_of_Man_ by Tom Paine (1791.)                    *
*****
```

Subject: Re: bug in IDL's hanning() window-generating function
Posted by [bennetsc](#) on Thu, 02 Aug 2001 07:16:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <3B681285.B12C4939@fz-juelich.de>,
Jaco van Gorkom <j.c.van.gorkom@fz-juelich.de> wrote:
> David Fanning wrote:
>> Scott Bennett writes after a long analysis of the Hanning function:
>>
>>> In any case, I think the point Harris made is that a discrete
>>> sampling of a window function should not taper to the same value at
>>> the end that it has at the beginning because to do so would include
>>> the first sample of the *next* period (windowed segment.) So IDL's
>>> hanning() gets it wrong for both even- and odd-length windows. :-(
>>
>> Uh, huh. And how did RSI respond when you contacted them
>> about it?
>
> I suspect they would suggest the following workaround:
>
> function Harris, n, _EXTRA=extra
> return, (hanning(n+1, _EXTRA=extra))[0:n-1]
> end
>
That looks like it should work for Hann, Hamming, and other

raised cosine windows. I am not prepared to vouch for that method for all windows.

> Scott, I've read your post, but I'm not sure I'm getting the point.
> Tapering from zero to zero seems like good idea to me, the symmetry
> sort of "feels good". Besides, it is convenient to have the weight of
> the window at its centre. What exactly is so wrong with it?
>

First, remember that a continuous, finite Fourier transform operates over a time (or space, but I'm just going to refer to time) domain of finite length T . The lowest frequency that can be resolved is the one whose period $1/f = T$; i.e. one full period takes exactly the length of the time over which the integral operates.

So now consider a periodic function at that frequency. Take, for example, a sine function like $y = \cos(2\pi t/T)$, where $0 \leq t \leq T$. Note that the function begins at $\cos(0)=1$. The assumption underlying the Fourier transform is that the basis functions are all periodic. So the finite Fourier transform is based upon the same assumption, but also that the function value over the time period from 0 to T exactly repeats ad infinitum. Therefore the time period is closed at the starting point and open at the ending. The open end is the point at which the cycle is assumed to begin its repetition of the starting point; i.e. it's the starting point of the next cycle or period, if another cycle were to exist.

The discrete, finite Fourier transform operates upon a sampling of the continuous function taken at N regular intervals of length Δt , so the samples are at $n * \Delta t$, where $n = 0, 1, 2, \dots, N-1$. Note that a sample taken at $t = N * \Delta t$ represents the point at which the cycle would repeat if more samples existed because the samples are taken at the start of each time interval, not at the end of each time interval. (A data window is supposed to have the same period as the data being windowed, so the data window's period likewise is closed at the left (starting) end and open-ended at the right.)

To see what happens when the discrete transform is applied to one too many samples, try the following. (Set color1 and color2 to vividly contrasting colors in your preferred color table.)

```
samples = cos(2. * !pi * findgen(9)/ 8.)
```

That gives us a sampled cosine function for one period of eight samples, plus an extra for our experiment. Next, get the discrete transforms and look at them.

```
gooddft = fft(samples[0:7])  
baddft = fft(samples)  
plot, float(gooddft), /nodata, yrange=[-.4, .6]  
oplot, float(gooddft), psym=1, color=color1  
oplot, imaginary(gooddft), psym=7, color=color2
```

The above should yield a graph with a color1 (real component) spike of .5 above the points for 1 and 7, corresponding to frequencies of $1/T$ and $-1/T$, and zeros everywhere else (within the limits of precision, etc.) and color2 (imaginary component) values of zero everywhere.

Then try:

```
plot, float(baddft), /nodata, yrange=[-.4, .6]
oplot, float(baddft), psym=1, color=color1
oplot, imaginary(baddft), psym=7, color=color2
```

This should give a graph showing a non-zero real (color1) mean value over the zero frequency, real (color1) peaks over the points 1 and 8, corresponding to frequencies of $1/T$ and $-1/T$, but a little higher than a value of .5 and imaginary (color2) points just less than .2 and just greater than -.2 over the same horizontal positions. Note that real and imaginary components of other frequencies are also non-zero. We know this cannot be correct because we specified the generating function to have only one frequency, which the transform splits in half over the symmetric frequencies of $1/T$ and $-1/T$, and we know that the mean of a cosine function over one complete period is zero. The amplitudes (both real and imaginary) of all other frequencies should be zero.

So a Hann window, and other windows I've seen so far, should not include a repetition of its first coefficient as the last coefficient.

Scott Bennett, Comm. ASMELG, CFIAG
College of Oceanic and Atmospheric

Sciences

Oregon State University
Corvallis, Oregon 97331

* Internet: sbennett at oce.orst.edu *

* "Lay then the axe to the root, and teach governments humanity. *

* It is their sanguinary punishments which corrupt mankind." *

* -- _The_Rights_of_Man_ by Tom Paine (1791.) *

Subject: Re: bug in IDL's hanning() window-generating function

Posted by [bennetsc](#) on Thu, 02 Aug 2001 07:21:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

In article <3B681285.B12C4939@fz-juelich.de>,
Jaco van Gorkom <j.c.van.gorkom@fz-juelich.de> wrote:
> David Fanning wrote:

```
>> Scott Bennett writes after a long analysis of the Hanning function:
>>
>>> In any case, I think the point Harris made is that a discrete
>>> sampling of a window function should not taper to the same value at
>>> the end that it has at the beginning because to do so would include
>>> the first sample of the *next* period (windowed segment.) So IDL's
>>> hanning() gets it wrong for both even- and odd-length windows. :-(
>>
>> Uh, huh. And how did RSI respond when you contacted them
>> about it?
>
> I suspect they would suggest the following workaround:
>
> function Harris, n, _EXTRA=extra
>   return, (hanning(n+1, _EXTRA=extra))[0:n-1]
> end
>
> That looks like it should work for Hann, Hamming, and other
> raised cosine windows. I am not prepared to vouch for that method
> for all windows.
```

```
> Scott, I've read your post, but I'm not sure I'm getting the point.
> Tapering from zero to zero seems like good idea to me, the symmetry
> sort of "feels good". Besides, it is convenient to have the weight of
> the window at its centre. What exactly is so wrong with it?
>
```

First, remember that a continuous, finite Fourier transform operates over a time (or space, but I'm just going to refer to time) domain of finite length T . The lowest frequency that can be resolved is the one whose period $1/f = T$; i.e. one full period takes exactly the length of the time over which the integral operates.

So now consider a periodic function at that frequency. Take, for example, a sine function like $y = \cos(2 \pi t/T)$, where $0 \leq t \leq T$. Note that the function begins at $\cos(0)=1$. The assumption underlying the Fourier transform is that the basis functions are all periodic. So the finite Fourier transform is based upon the same assumption, but also that the function value over the time period from 0 to T exactly repeats ad infinitum. Therefore the time period is closed at the starting point and open at the ending. The open end is the point at which the cycle is assumed to begin its repetition of the starting point; i.e. it's the starting point of the next cycle or period, if another cycle were to exist.

The discrete, finite Fourier transform operates upon a sampling of the continuous function taken at N regular intervals of length Δt , so the samples are at $n * \Delta t$, where $n = 0, 1, 2, \dots, N-1$. Note that a sample taken at $t = N * \Delta t$ represents the point at which the cycle would repeat if more samples existed because the samples are taken at the start of each time interval, not at the end

of each time interval. (A data window is supposed to have the same period as the data being windowed, so the data window's period likewise is closed at the left (starting) end and open-ended at the right.)

To see what happens when the discrete transform is applied to one too many samples, try the following. (Set color1 and color2 to vividly contrasting colors in your preferred color table.)

```
samples = cos(2. * !pi * findgen(9)/ 8.)
```

That gives us a sampled cosine function for one period of eight samples, plus an extra for our experiment. Next, get the discrete transforms and look at them.

```
gooddft = fft(samples[0:7])
baddft = fft(samples)
plot, float(gooddft), /nodata, yrange=[-.4, .6]
oplot, float(gooddft), psym=1, color=color1
oplot, imaginary(gooddft), psym=7, color=color2
```

The above should yield a graph with a color1 (real component) spike of .5 above the points for 1 and 7, corresponding to frequencies of $1/T$ and $-1/T$, and zeros everywhere else (within the limits of precision, etc.,) and color2 (imaginary component) values of zero everywhere.

Then try:

```
plot, float(baddft), /nodata, yrange=[-.4, .6]
oplot, float(baddft), psym=1, color=color1
oplot, imaginary(baddft), psym=7, color=color2
```

This should give a graph showing a non-zero real (color1) mean value over the zero frequency, real (color1) peaks over the points 1 and 8, corresponding to frequencies of $1/T$ and $-1/T$, but a little higher than a value of .5 and imaginary (color2) points just less than .2 and just greater than -.2 over the same horizontal positions. Note that real and imaginary components of other frequencies are also non-zero. We know this cannot be correct because we specified the generating function to have only one frequency, which the transform splits in half over the symmetric frequencies of $1/T$ and $-1/T$, and we know that the mean of a cosine function over one complete period is zero. The amplitudes (both real and imaginary) of all other frequencies should be zero.

So a Hann window, and other windows I've seen so far, should not include a repetition of its first coefficient as the last coefficient.

Scott Bennett, Comm. ASME LG, CFIAG
College of Oceanic and Atmospheric

Sciences

Oregon State University
Corvallis, Oregon 97331

```
* Internet:      sbennett at oce.orst.edu      *
* -----*
* "Lay then the axe to the root, and teach governments humanity.  *
* It is their sanguinary punishments which corrupt mankind."      *
* -- _The_Rights_of_Man_ by Tom Paine (1791.)                      *
*****
```

Subject: Re: bug in IDL's hanning() window-generating function
Posted by [Clay Kirkendall](#) on Thu, 02 Aug 2001 13:35:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

There is no doubt that the IDL supplied hanning window is symmetric but not "fft-symmetric" as it should be for fft processing. I would guess that for most applications this difference will not significantly impact the results, but for situations where there is a large signal to noise ratio (> 80 dB) with a sinusoidal signal exactly centered in an fft bin there can be major differences between the two windows. Most real world problems (at least that I have come across) dont meet this situation and the observed differences are minor. Of course there is no reason to use the wrong window when the correct one is easily generated.

Clay

bennetsc@NOSPAMucs.orst.edu wrote:

```
> In article <3B681285.B12C4939@fz-juelich.de>,
> Jaco van Gorkom <j.c.van.gorkom@fz-juelich.de> wrote:
>> David Fanning wrote:
>>> Scott Bennett writes after a long analysis of the Hanning function:
>>>
>>>> In any case, I think the point Harris made is that a discrete
>>>> sampling of a window function should not taper to the same value at
>>>> the end that it has at the beginning because to do so would include
>>>> the first sample of the *next* period (windowed segment.) So IDL's
>>>> hanning() gets it wrong for both even- and odd-length windows. :-(
>>>
>>> Uh, huh. And how did RSI respond when you contacted them
>>> about it?
>>
>> I suspect they would suggest the following workaround:
>>
>> function Harris, n, _EXTRA=extra
>>   return, (hanning(n+1, _EXTRA=extra))[0:n-1]
>> end
```



```

>>
> That looks like it should work for Hann, Hamming, and other
> raised cosine windows. I am not prepared to vouch for that method
> for all windows.
>
>> Scott, I've read your post, but I'm not sure I'm getting the point.
>> Tapering from zero to zero seems like good idea to me, the symmetry
>> sort of "feels good". Besides, it is convenient to have the weight of
>> the window at its centre. What exactly is so wrong with it?
>>
> First, remember that a continuous, finite Fourier transform
> operates over a time (or space, but I'm just going to refer to time)
> domain of finite length T. The lowest frequency that can be resolved
> is the one whose period  $1/f = T$ ; i.e. one full period takes exactly
> the length of the time over which the integral operates.
> So now consider a periodic function at that frequency. Take,
> for example, a sine function like  $y = \cos(2 \pi t/T)$ , where  $0 \leq t \leq T$ .
> Note that the function begins at  $\cos(0)=1$ . The assumption underlying
> the Fourier transform is that the basis functions are all periodic.
> So the finite Fourier transform is based upon the same assumption, but
> also that the function value over the time period from 0 to T exactly
> repeats ad infinitum. Therefore the time period is closed at the
> starting point and open at the ending. The open end is the point at
> which the cycle is assumed to begin its repetition of the starting
> point; i.e. it's the starting point of the next cycle or period, if
> another cycle were to exist.
> The discrete, finite Fourier transform operates upon a sampling
> of the continuous function taken at N regular intervals of length
>  $\Delta t$ , so the samples are at  $n * \Delta t$ , where  $n = 0, 1, 2, \dots, N-1$ .
> Note that a sample taken at  $t = N * \Delta t$  represents the point at
> which the cycle would repeat if more samples existed because the
> samples are taken at the start of each time interval, not at the end
> of each time interval. (A data window is supposed to have the same
> period as the data being windowed, so the data window's period likewise
> is closed at the left (starting) end and open-ended at the right.)
> To see what happens when the discrete transform is applied to one
> too many samples, try the following. (Set color1 and color2 to vividly
> contrasting colors in your preferred color table.)
>
> samples = cos(2. * !pi * findgen(9)/ 8.)
>
> That gives us a sampled cosine function for one period of eight
> samples, plus an extra for our experiment. Next, get the discrete
> transforms and look at them.
>
> gooddft = fft(samples[0:7])
> baddft = fft(samples)
> plot, float(gooddft), /nodata, yrange=[-.4, .6]

```



```

> oplot, float(gooddft), psym=1, color=color1
> oplot, imaginary(gooddft), psym=7, color=color2
>
> The above should yield a graph with a color1 (real component) spike of
> .5 above the points for 1 and 7, corresponding to frequencies of 1/T
> and -1/T, and zeros everywhere else (within the limits of precision,
> etc.,) and color2 (imaginary component) values of zero everywhere.
> Then try:
>
> plot, float(baddft), /nodata, yrange=[-.4, .6]
> oplot, float(baddft), psym=1, color=color1
> oplot, imaginary(baddft), psym=7, color=color2
>
> This should give a graph showing a non-zero real (color1) mean value
> over the zero frequency, real (color1) peaks over the points 1 and 8,
> corresponding to frequencies of 1/T and -1/T, but a little higher than
> a value of .5 and imaginary (color2) points just less than .2 and just
> greater than -.2 over the same horizontal positions. Note that real
> and imaginary components of other frequencies are also non-zero. We
> know this cannot be correct because we specified the generating
> function to have only one frequency, which the transform splits in
> half over the symmetric frequencies of 1/T and -1/T, and we know that
> the mean of a cosine function over one complete period is zero. The
> amplitudes (both real and imaginary) of all other frequencies should
> be zero.
> So a Hann window, and other windows I've seen so far, should not
> include a repetition of its first coefficient as the last coefficient.
>
> Scott Bennett, Comm. ASME LG, CFIAG
> College of Oceanic and Atmospheric
> Sciences
> Oregon State University
> Corvallis, Oregon 97331
> *****
> * Internet: sbennett at oce.orst.edu *
> * -----*
> * "Lay then the axe to the root, and teach governments humanity. *
> * It is their sanguinary punishments which corrupt mankind." *
> * -- _The_Rights_of_Man_ by Tom Paine (1791.) *
> *****

```

Subject: Re: bug in IDL's hanning() window-generating function
 Posted by [Paul van Delst](#) on Thu, 02 Aug 2001 15:42:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

bennetsc@NOSPAMucs.orst.edu wrote:
 >

> In article <MPG.15d1a1e0a012bdcd989e44@news.frii.com>,
> David Fanning <david@dfanning.com> wrote:
>> Scott Bennett writes after a long analysis of the Hanning function:
>>
>>> In any case, I think the point Harris made is that a discrete
>>> sampling of a window function should not taper to the same value at
>>> the end that it has at the beginning because to do so would include
>>> the first sample of the *next* period (windowed segment.) So IDL's
>>> hanning() gets it wrong for both even- and odd-length windows. :-(
>>
>> Uh, huh. And how did RSI respond when you contacted them
>> about it?
>>
> I didn't contact them. I'm disgusted with them after seeing how
> they cheated my faculty advisor and, when I asked them to rectify
> their errors, they refused to do so. I recommended to my advisor that
> he turn the whole matter over to the university's legal counsel. He
> told me that he was not going to renew the service contract any more,
> so I doubt he will bother with having the legal counsel deal with it,
> but either way it will be his decision and it's now out of my hands.
> Given their denial of service while we had a service contract, I doubt
> they would do any better now that we no longer have a service contract.

Oi vey. another future matlab user.... ? :o)

--

Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
 Alexander Pope.

Subject: Re: bug in IDL's hanning() window-generating function
Posted by [Clay Kirkendall](#) on Thu, 02 Aug 2001 17:53:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

FYI: the hanning() routine that I have from matlab (ver 1.4, 1994) is in worse shape than the one provided with IDL. It is symmetric (not fft-symmetric) and it does not even taper to zero.

Clay

Paul van Delst wrote:

> bennetsc@NOSPAMucs.orst.edu wrote:
>>
>> In article <MPG.15d1a1e0a012bdcd989e44@news.frii.com>,

>> David Fanning <david@dfanning.com> wrote:
>>> Scott Bennett writes after a long analysis of the Hanning function:
>>>
>>>> In any case, I think the point Harris made is that a discrete
>>>> sampling of a window function should not taper to the same value at
>>>> the end that it has at the beginning because to do so would include
>>>> the first sample of the *next* period (windowed segment.) So IDL's
>>>> hanning() gets it wrong for both even- and odd-length windows. :-(
>>>
>>> Uh, huh. And how did RSI respond when you contacted them
>>> about it?
>>>
>> I didn't contact them. I'm disgusted with them after seeing how
>> they cheated my faculty advisor and, when I asked them to rectify
>> their errors, they refused to do so. I recommended to my advisor that
>> he turn the whole matter over to the university's legal counsel. He
>> told me that he was not going to renew the service contract any more,
>> so I doubt he will bother with having the legal counsel deal with it,
>> but either way it will be his decision and it's now out of my hands.
>> Given their denial of service while we had a service contract, I doubt
>> they would do any better now that we no longer have a service contract.
>
> Oi vey. another future matlab user.... ? :o)
>
> --
> Paul van Delst A little learning is a dangerous thing;
> CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
> Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
> Fax:(301)763-8545 And drinking largely sobers us again.
> Alexander Pope.

Subject: Re: bug in IDL's hanning() window-generating function
Posted by [Pavel A. Romashkin](#) on Thu, 02 Aug 2001 21:30:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

bennetsc@NOSPAMucs.orst.edu wrote:

>
> I didn't contact them. I'm disgusted with them after seeing how
> they cheated my faculty advisor and, when I asked them to rectify
> their errors, they refused to do so.

Wow. Are you sure it was not a misunderstanding? I have never had a
problem on a consumer level with RSI. Of course, I am not trying to
sparkle an outpour of complains by writing this.

Cheers,
Pavel

Subject: Re: bug in IDL's hanning() window-generating function
Posted by [bennetsc](#) on Fri, 03 Aug 2001 01:00:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <3B6974E4.CBCC8A96@noaa.gov>,
Paul van Delst <paul.vandelst@noaa.gov> wrote:
> bennetsc@NOSPAMucs.orst.edu wrote:
>>
>> In article <MPG.15d1a1e0a012bdcd989e44@news.frii.com>,
>> David Fanning <david@dfanning.com> wrote:
>>> Scott Bennett writes after a long analysis of the Hanning function:
>>>
>>>> In any case, I think the point Harris made is that a discrete
>>>> sampling of a window function should not taper to the same value at
>>>> the end that it has at the beginning because to do so would include
>>>> the first sample of the *next* period (windowed segment.) So IDL's
>>>> hanning() gets it wrong for both even- and odd-length windows. :-(
>>>
>>> Uh, huh. And how did RSI respond when you contacted them
>>> about it?
>>>
>> I didn't contact them. I'm disgusted with them after seeing how
>> they cheated my faculty advisor and, when I asked them to rectify
>> their errors, they refused to do so. I recommended to my advisor that
>> he turn the whole matter over to the university's legal counsel. He
>> told me that he was not going to renew the service contract any more,
>> so I doubt he will bother with having the legal counsel deal with it,
>> but either way it will be his decision and it's now out of my hands.
>> Given their denial of service while we had a service contract, I doubt
>> they would do any better now that we no longer have a service
> contract.
>
> Oi vey. another future matlab user.... ? :o)
>
Who knows? It will depend upon which package, if any, my next
employer wants me to use. I don't even know what my chances are
of ever being involved in time series data analysis work again after
I finish this degree program. (I suppose, if I can find a suitable
disk drive to replace the one that died on my ancient NeXT, I could
mess around with Mathematica 3.02 at home... :-)

Sciences
Scott Bennett, Comm. ASME LG, CFIAG
College of Oceanic and Atmospheric
Oregon State University
Corvallis, Oregon 97331

* Internet: sbennett at oce.orst.edu

*

* "Lay then the axe to the root, and teach governments humanity. *

* It is their sanguinary punishments which corrupt mankind." *

* -- _The_Rights_of_Man_ by Tom Paine (1791.) *

Subject: Re: bug in IDL's hanning() window-generating function
Posted by [bennetsc](#) on Fri, 03 Aug 2001 01:42:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <3B69C65C.74C8539D@noaa.gov>,
Pavel A. Romashkin <pavel.romashkin@noaa.gov> wrote:
> bennetsc@NOSPAMucs.orst.edu wrote:

>>
>> I didn't contact them. I'm disgusted with them after seeing how
>> they cheated my faculty advisor and, when I asked them to rectify
>> their errors, they refused to do so.
>
> Wow. Are you sure it was not a misunderstanding? I have never had a
> problem on a consumer level with RSI. Of course, I am not trying to
> sparkle an outpour of complains by writing this.

My advisor got a couple of IDL licenses back at IDL 5.0, one for his Mac and one for a Sun, which is the one I use. After 5.1 came out, the only way we found out about it was when we heard and read that other sites had gotten their updates and were using them. Later, we got 5.2, which came as a CDROM and a small book titled _What's New_in_IDL_5.2?_. However, that book only covered changes between 5.1 and 5.2, so I didn't have what I needed to start using 5.2 if it were installed. Eventually, we heard that other people had gotten their 5.3 updates, but we never got one of those either. Other people in our college also have IDL licenses, so I was able to get the college's computer/networking support folks to install 5.3 and we did get a license string from RSI for it. However, without up-to-date documentation, especially on changes all the way from 5.0 to 5.3, it was not practical to proceed from 5.0.

I don't recall whether I had contacted RSI before this time about missing releases and lack of documentation, but at this point I certainly did. My first attempt was ignored. After waiting more than adequate time for response from corporate staff that might be backlogged, I tried again and got a response from a "Technical Support Engineer," who claimed to have forwarded my message to our "maintenance rep" and also claimed there would be a charge if we wanted the manuals we were already supposed to have gotten. (She didn't mention the CDROMs.) No word was ever received from the "maintenance rep." What I needed most by this time was an up-to-date version of the _Using_IDL_ book, but probably would have managed fine with a complete sequence

of the _What's_New..._ books. She suggested using the on-line "help" facility and otherwise gave me the brush-off.

I was and still am quite busy with my work and dealing with unresponsive software companies isn't really supposed to be taking up my time, so I did nothing to bring more frustration upon myself from RSI for some time after that. Eventually, we got a CDROM for 5.4 and a book titled _What's_New_in_IDL_5.4_, which, again, was insufficient for my needs. So I tried contacting RSI again. This time a different person responded and, naturally, knew nothing about the situation, so I started all over. She told me that they stopped shipping new manuals to educational licensees when they started giving a greatly increased educational discount. (I don't know when that was supposed to have happened.) If we wanted new manuals, they would cost us. So I told her to just ship us the update packages that they had failed to send us and I would use the _What's_New..._ books to get up to date. She said all she had for 5.1 and 5.3 were the CDROMs, not the books that were shipped with them. I said they should make good on the service contract by providing us an adequate substitute free of charge for what they had failed to ship us in the first place. She said they would not do that, but said they keep a pile of used manuals that customers return for some reason (abandonment of package?) and that she would look for some old ones to send us.

That was the last I heard from her until today. Apparently, she saw or was given a copy of the complaint I posted last night. She pretended in today's note not to know what my complaint was about, nor the name of the licensee (i.e. my advisor.) So to answer Pavel's question, it does not seem to me to be conceivable that there was any misunderstanding. They simply cheated my advisor and refused to make good when confronted with the facts.

Scott Bennett, Comm. ASME LG, CFIAG
College of Oceanic and Atmospheric

Sciences

Oregon State University
Corvallis, Oregon 97331

```
* Internet:      sbennett at oce.orst.edu      *
*-----*
* "Lay then the axe to the root, and teach governments humanity.      *
* It is their sanguinary punishments which corrupt mankind."      *
* -- _The_Rights_of_Man_ by Tom Paine (1791.)      *
```

Subject: Re: bug in IDL's hanning() window-generating function
Posted by [Kenneth P. Bowman](#) on Fri, 03 Aug 2001 02:32:30 GMT

Without making any comment on your problems with RSI, I can say that at least the 5.3 and 5.4 releases (and possibly earlier?) have included all the IDL documentation in PDF format, which in my experience is vastly superior to either the paper manuals or the old help system.

Also, you can always download the latest version of IDL (including the documentation) from the RSI web site. We have always found it easy to install updates.

Ken Bowman

Subject: Re: bug in IDL's hanning() window-generating function
Posted by [david\[2\]](#) on Fri, 03 Aug 2001 03:31:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

Scott Bennett writes:

- > They simply cheated my advisor and
- > refused to make good when confronted with the facts.

I don't know. Pretty strong words for what appears from the facts you present to be a misunderstanding to me. But I know from personal experience that educational institutions are extremely difficult to work with. The left hand never seems to know what the right hand is doing. The people in charge of purchasing the software never seem to have a clue who is using it. Hence, information like where the manuals are and what the terms of the deal are are forever being lost.

And the people at educational institutions can be difficult.

I'm still livid over a deal I did at a large university in Arizona. They wanted cheap software. I went to bat for them, stuck my neck WAY out to make it happen, and then found myself accused in this newsgroup of being a "software junkie", since I had clearly lowered the price to get them "hooked" on the software. Scheesh! After that experience I spent a couple of years saying to hell with educational institutions.

It goes both ways.

Cheers,

David

P.S. Let's just say having the manuals probably wouldn't have helped as much as you think. You would be better off with a good book. There are several now.

--

David Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438 E-Mail: davidf@dfanning.com

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: bug in IDL's hanning() window-generating function

Posted by [bennetsc](#) on Sat, 04 Aug 2001 05:26:27 GMT

[View Forum Message](#) <> [Reply to Message](#)

In article <020820012132307408%kbowman@null.net>,
Kenneth P. Bowman <kbowman@null.net> wrote:

>

> Without making any comment on your problems with RSI, I can say that at
> least the 5.3 and 5.4 releases (and possibly earlier?) have included
> all the IDL documentation in PDF format, which in my experience is
> vastly superior to either the paper manuals or the old help system.

I looked at the 5.3 stuff in PDF format, but my recollection is that it was broken up into lots of very small files, very inconvenient and expensive to print hardcopy manuals, which then would have had to be punched and put into binders in order to be usable. Since we were supposed to have received the update information anyway, it's not a grad student's place to squander resources on it.

The IDL 5.0 help system is nearly worthless. Perhaps the later versions' help systems are better, but they would still be a grossly inefficient and frustrating way of finding out what had changed from version to version.

>

> Also, you can always download the latest version of IDL (including the
> documentation) from the RSI web site. We have always found it easy to
> install updates.

>

I'm a grad student here. Installation of large software packages with licensing schemes is reserved to the college's computer support staff.

FWIW, I would like someday, preferably in the very near future, to finish this degree. It's a bit late in the process for me to devote much effort to switching releases at this point.

Sciences
Scott Bennett, Comm. ASMELG, CFIAG
College of Oceanic and Atmospheric
Oregon State University
Corvallis, Oregon 97331

```
*****
* Internet:      sbennett at oce.orst.edu          *
* -----*
* "Lay then the axe to the root, and teach governments humanity.  *
* It is their sanguinary punishments which corrupt mankind."      *
* -- _The_Rights_of_Man_ by Tom Paine (1791.)                    *
*****
```

Subject: Re: bug in IDL's hanning() window-generating function
Posted by [Jaco van Gorkom](#) on Fri, 10 Aug 2001 16:16:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

bennetsc@NOSPAMucs.orst.edu wrote:

> Jaco van Gorkom <j.c.van.gorkom@fz-juelich.de> wrote:

```
>> function Harris, n, _EXTRA=extra
>>   return, (hanning(n+1, _EXTRA=extra))[0:n-1]
>> end
>>
>   That looks like it should work for Hann, Hamming, and other
> raised cosine windows. I am not prepared to vouch for that method
> for all windows.
```

Well, I inherited *some* of the functionality from the built-in HANNING() function. Therefore this is not very likely to produce other types of windows, symmetric or not.

> So a Hann window, and other windows I've seen so far, should not
> include a repetition of its first coefficient as the last coefficient.

Thanks to all for the explanations and examples. I am now definitely a convert to 5.5-style hanning windows. Even though I will certainly never encounter an 80 dB signal-to-noise ratio...

Jaco
