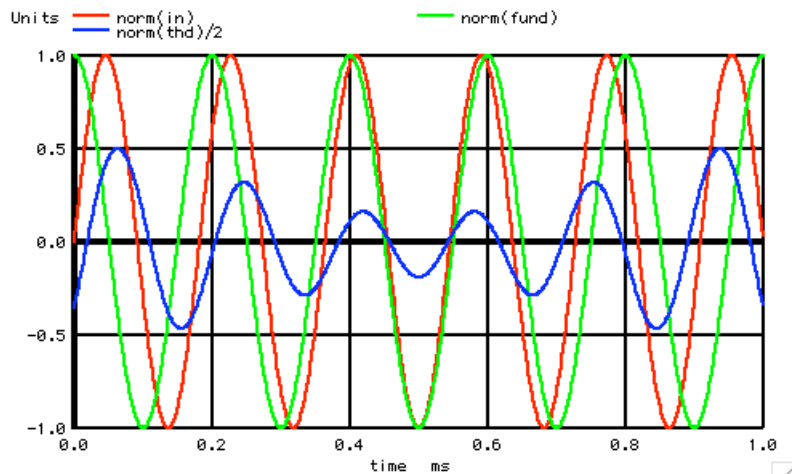


***=====FFT LEAKAGE=====**

**HAVING TWO WAY TRANSLATION BETWEEN TIME AND SPECTRUM,
THE FFT LEAKAGE WAVEFORM CAN NOW BE LOOKED AT
WHEN NOT USING FFT WINDOWING.**



**Does spectrum leakage always look like AM and PM
of the fundamental? Frequency is now begin defined
with a DC voltage source.**

```

=====
FFT_Leakage_tests
*=====Create_Signal=====
VTime      VTime 0      DC    0      PWL( 0 0 1 1)
Vfreq      Vfreq 0      DC    5.001k
BVAC       IN    0      V =   sin( 6.283185307179586*V(VFreq)*V(VTime))
.control
*TRAN      TSTEP  TSTOP  TSTART TMAX  ?UIC?
tran       1u    .999m  0      1u
set        pensize = 2
linearize
let        numb2 = length(in)
print     numb2

```

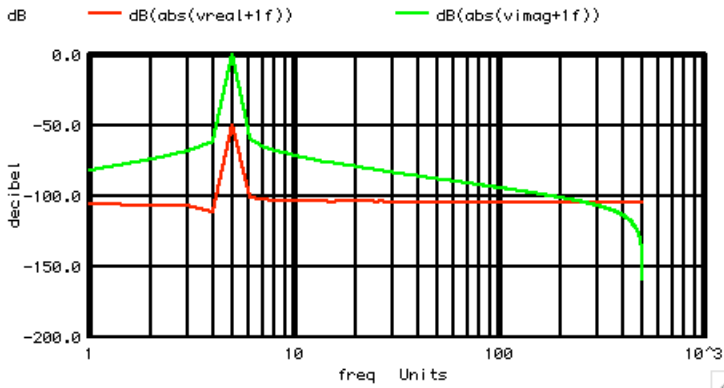
***=====FFT_and_Translate_into_dB_freq=====**

This example uses only 1000 points.

```

=====
let        ac      = in +j(0)
let        ac_fft  = fft(ac)
let        numb_f2 = (numb2)/2 -1
compose    freq_   start = 1 stop = $&numb_f2 step =1
compose    vreal   start = 1 stop = $&numb_f2 step =1
compose    vimag   start = 1 stop = $&numb_f2 step =1
let        i       = 0
repeat    $&numb_f2
let        freq[i] = freq[i]
let        vreal[i] = 2*real(ac_fft[i+1])
let        vimag[i] = 2*imag(ac_fft[i+1])
let        i       = i +1
end
plot      dB(abs(vreal+1f)) dB(abs(vimag+1f)) vs freq xlog

```



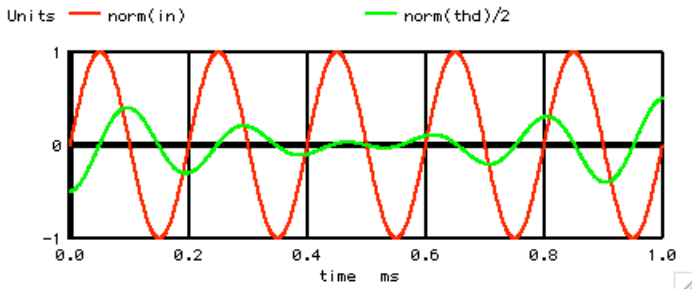
*=====View_The_THD=====

Now subtract a constant 5KHz term to view the distortion using the norm function.

```

=====
let      funBin          = 5k/1000
let      unvect          = unitvec($&numb2)
let      fundspec        = unvect*0 +j(0)
let      fundspec[funBin] = real(ac_fft[funBin])      +j(imag(ac_fft[funBin] ))
let      fundspec[numb2-funBin] = real(ac_fft[numb2-funBin]) +j(imag(ac_fft[numb2-funBin] ))
let      fund            = ifft(fundspec)
let      dc_offset       = real(ac_fft[0])
let      thdspec         = ac_fft
let      thdspec[0]      = 0          +j(0)
let      thdspec[funBin] = 0          +j(0)
let      thdspec[numb2-funBin] = 0      +j(0)
let      thd             = ifft(thdspec)
plot     norm(in) norm(thd)/2
=====

```



*=====And_the_values_can_be_calculated.=====

The resulting table shows the frequency as well as THD.

```

=====
let      rms_Fund        = sqrt(mean(fund*fund))
let      rms_THD         = sqrt(mean(thd*thd))
let      THD_percent     = 100*rms_THD/rms_Fund
let      FREQ_Hz        = VFreq[0]
echo     "Freq_Hz=$&FREQ_Hz THD_percent=$&THD_percent Fund_rms=$&rms_Fund THD_rms=$&rms_THD"
=====

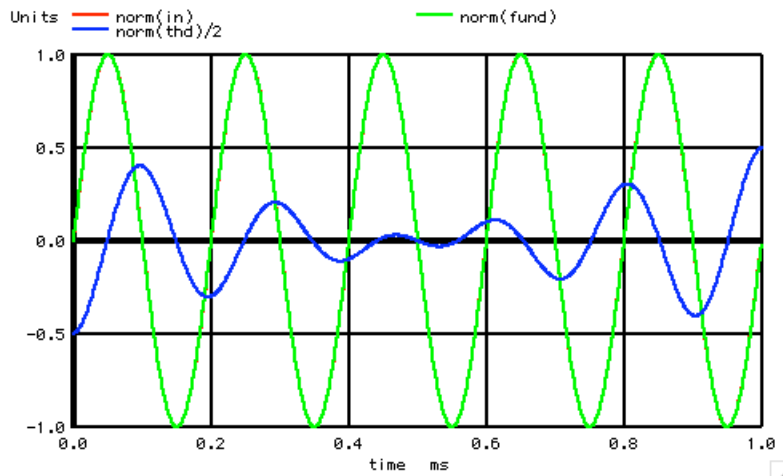
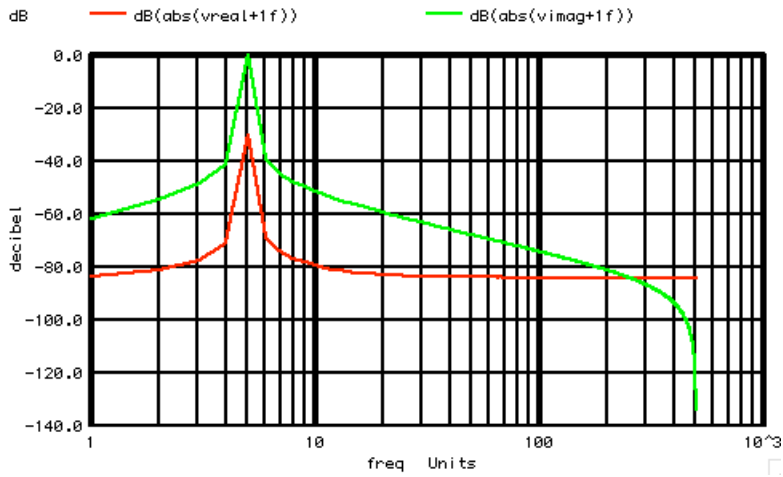
```

Freq_Hz=5001 THD_percent=0.18168 Fund_rms=0.7073 THD_rms=0.00128

*=====Add_another_Order_of_Magnitude=====

Vfreq Vfreq 0 DC 5.01k

=====

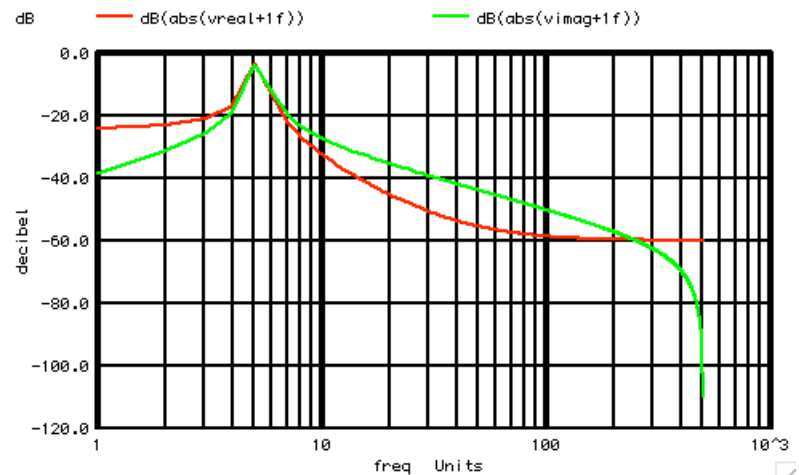


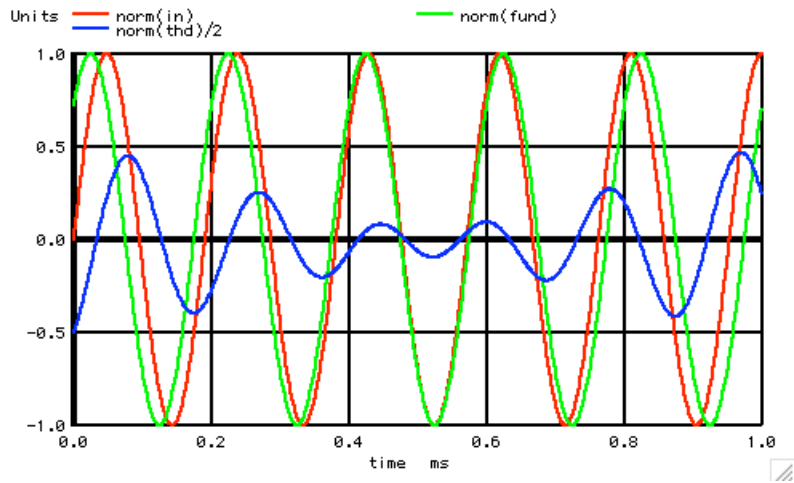
Freq_Hz=5010 THD_percent=1.82034 Fund_rms=0.706626 THD_rms=0.012863

The THD still looks like it is being amplitude and phase modulated in the same way. Only the magnitude appears to be increasing.

***=====Now_Go_to_5.25KHz=====**

Vfreq Vfreq 0 DC 5.25k





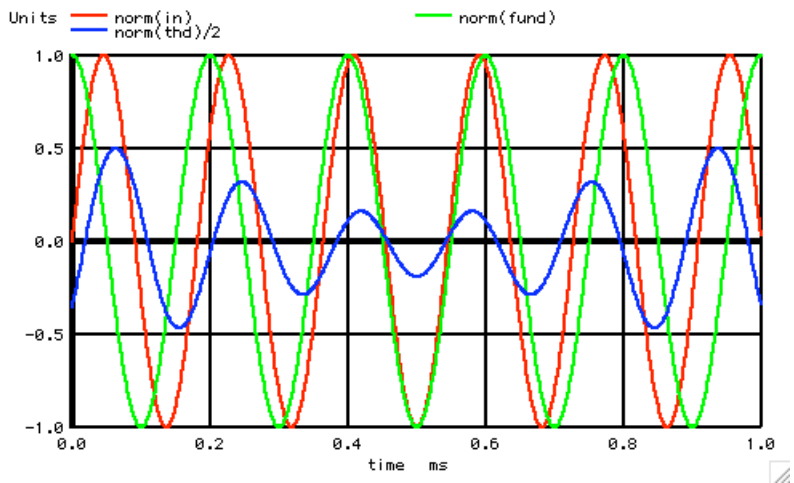
Freq_Hz=5250 THD_percent=48.1168 Fund_rms=0.636635 THD_rms=0.306328

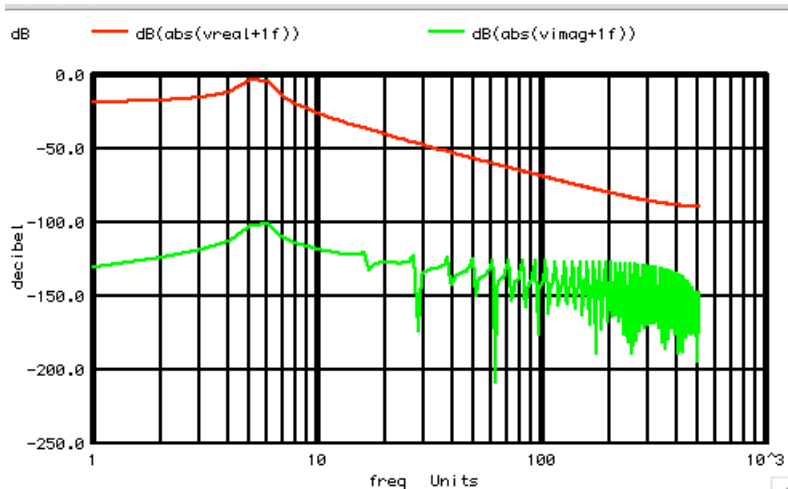
The start and stop phase of the THD appears to be moving slightly at this level of frequency offset.

*=====Now_Go_to_5.5KHz=====

Vfreq Vfreq 0 DC 5.25k

=====





Freq_Hz=5500 THD_percent=111.05 Fund_rms=0.471861 THD_rms=0.524001

There is discontinuity in the start and stop phase of the THD signal. One would expect that to be the thing creating the spread out of the spectrum. The windowing functions all attenuate these endpoints where the discontinuities are taking place.

=====**Full_Netlist_For_Copy_Paste**=====

```

FFT_Leakage_tests
*=====Create_Signal=====
VTime      VTime 0      DC      0      PWL( 0      0      1      1)
Vfreq      Vfreq 0      DC      5.05k
BVAC       IN      0      V      =      sin( 6.283185307179586*V(VFreq)*V(VTime))
.control
*TRAN      TSTEP  TSTOP  TSTART TMAX   ?UIC?
tran       1u      .999m  0      1u
set        pensize = 2
linearize
let        numb2 = length(in)
print     numb2

*=====Do FFT and Plot As dB Freq=====
let        ac = in +j(0)
let        ac_fft=fft(ac)
let        numb_f2 = (numb2)/2 -1
compose    freq   start = 1 stop = $&numb_f2 step =1
compose    vreal  start = 1 stop = $&numb_f2 step =1
compose    vimag  start = 1 stop = $&numb_f2 step =1
let        i = 0
repeat     $&numb_f2
let        freq[i] = freq[i]
let        vreal[i] = 2*real(ac_fft[i+1])
let        vimag[i] = 2*imag(ac_fft[i+1])
let        i = i +1
end
plot      dB(abs(vreal+1f)) dB(abs(vimag+1f)) vs freq xlog

*=====Extract_Error_Signal=====
let        funBin      = 5k/1000
let        unvect      = unitvec($&numb2)
let        fundspec    = unvect*0 +j(0)
let        fundspec[funBin] = real(ac_fft[funBin]) +j(imag(ac_fft[funBin] ))
let        fundspec[numb2-funBin] = real(ac_fft[numb2-funBin]) +j(imag(ac_fft[numb2-funBin] ))
let        fund        = ifft(fundspec)
let        dc_offset  = real(ac_fft[0])
let        thdspec    = ac_fft
let        thdspec[0] = 0 +j(0)
let        thdspec[funBin] = 0 +j(0)
let        thdspec[numb2-funBin] = 0 +j(0)
let        thd        = ifft(thdspec)
plot      norm(in) norm(thd)/2

*=====Calc Values=====
let        rms_Fund    = sqrt(mean(fund*fund))
let        rms_THD    = sqrt(mean(thd*thd))
let        THD_percent = 100*rms_THD/rms_Fund
let        FREQ_Hz    = VFreq[0]
echo      "Freq_Hz=$&FREQ_Hz THD_percent=$&THD_percent Fund_rms=$&rms_Fund THD_rms=$&rms_THD "

```

.endc
.end