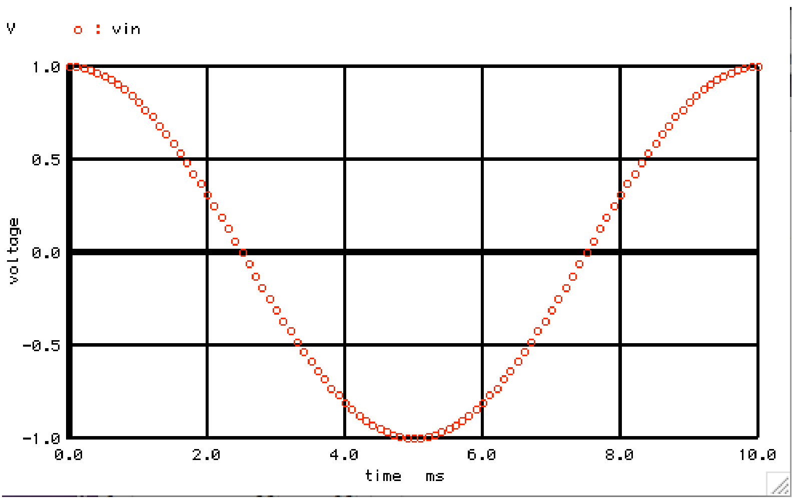
*=========FFT_SINE================

A sine wave can show the limits to the precision. This waveform is created to also check out the cos function.

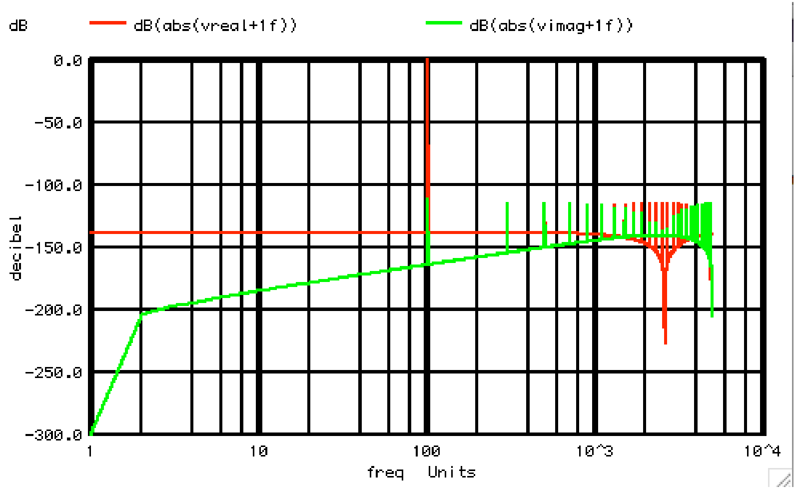
```
==================================================================
FFT_tests_SINE
VPI        VP      0      DC      3.141592653589793
V2PI       V2P     0      DC      6.283185307179586
VT         Vtime   0      DC      0       PWL ( 0 0 1 1 )
VFreq      VFreq   0      DC      100
BSIN       VIN     0      V =     cos(6.2831853072*v(VFreq)*v(Vtime) )
*BSIN      VIN     0      V =     cos(6.2831*v(VFreq)*v(Vtime) )
.control
*TRAN      TSTEP   TSTOP  TSTART TMAX
tran      .1m     .9999  0       .1m
set        pensize = 2
linearize
plot       vin  xlimit 0 10m pointplot
```



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

This appears to be about the best the spectrum output can get.

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

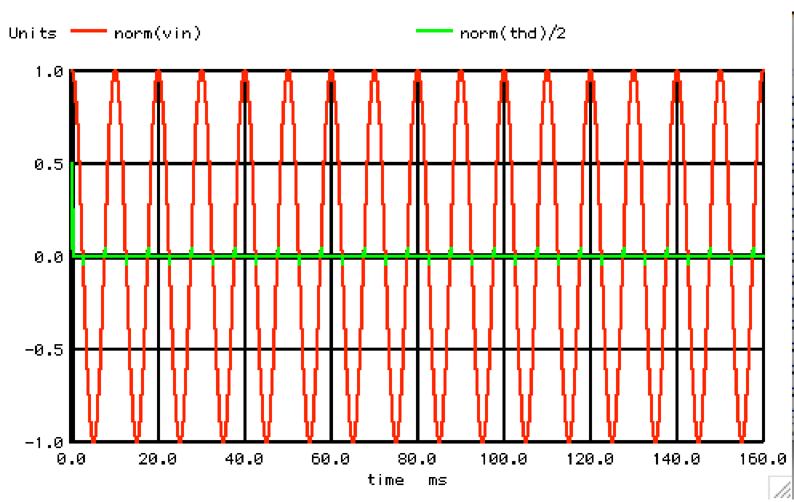


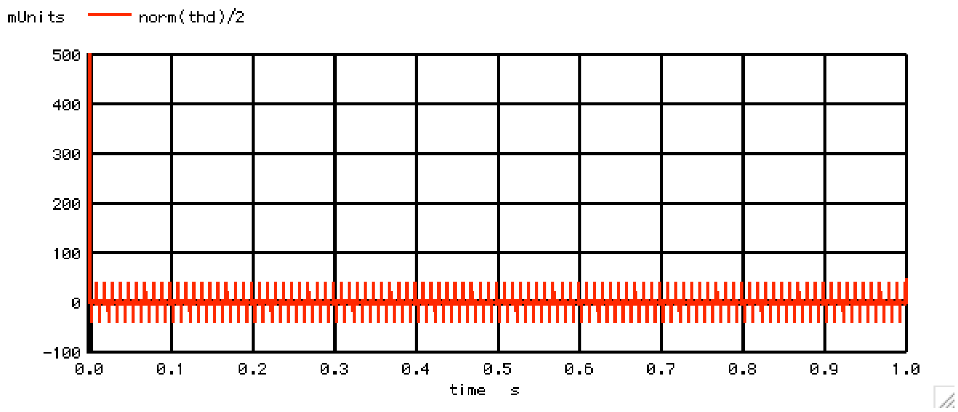
*=========View_The_THD====================

Now subtract the fundamental to view the distortion using the norm function.

==

```
let       sigAt                  = VFreq[0]
let       unvect                 = unitvec($&numb2)
let       fundspec               = unvect*0 +j(0)
let       fundspec[sigAt]        = real(ac_fft[sigAt])        +j(imag(ac_fft[sigAt] ))
let       fundspec[numb2-sigAt]  = real(ac_fft[numb2-sigAt])  +j(imag(ac_fft[numb2-sigAt] ))
let       fund                   = ifft(fundspec)
let       dc_ofset               = real(ac_fft[0])
let       thdspec                = ac_fft
let       thdspec[0]             = 0         +j(0)
let       thdspec[sigAt]         = 0         +j(0)
let       thdspec[numb2-sigAt]   = 0         +j(0)
let       thd                    = ifft(thdspec)
plot      norm(vin) norm(thd)/2    xlimit 0 .15
plot      norm(vin) norm(thd)/2    xlimit .84 .9999
plot      norm(thd)/2
```





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

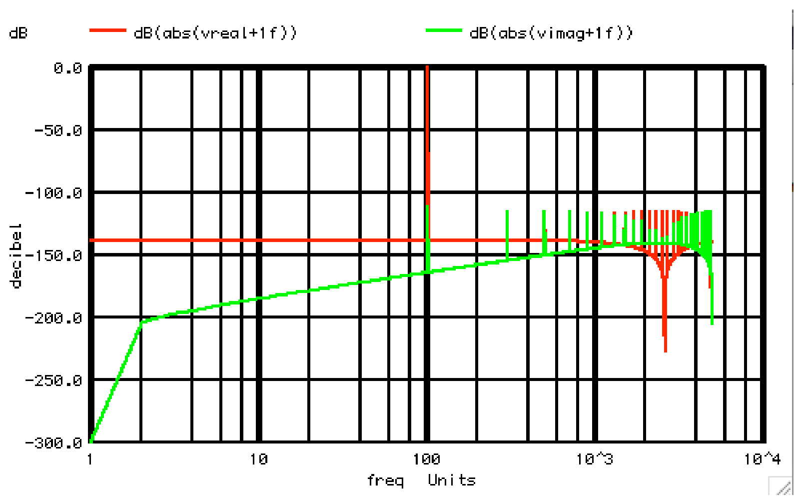
There appears to be some glitches in the THD output. The following prints out the THD when the signal is defined as such.

```
BSIN      VIN    0      V =     cos(6.2831853072*v(VFreq)*v(Vtime) )
```

```
==============================================================
let        rms_Fund                   = sqrt(mean(fund*fund))
let        rms_THD                    = sqrt(mean(thd*thd))
let        THD_percent                = 100*rms_THD/rms_Fund
echo       "THD_%=$&THD_percent OutDC=$&dc_ofset Fund_rms=$&rms_Fund THD_rms=$&rms_THD   "
==============================================================
 THD_%=0.00112717   Fund_rms=0.70707  THD_rms=7.9699E-06
```

*=========The_Distortion_checks_out===================

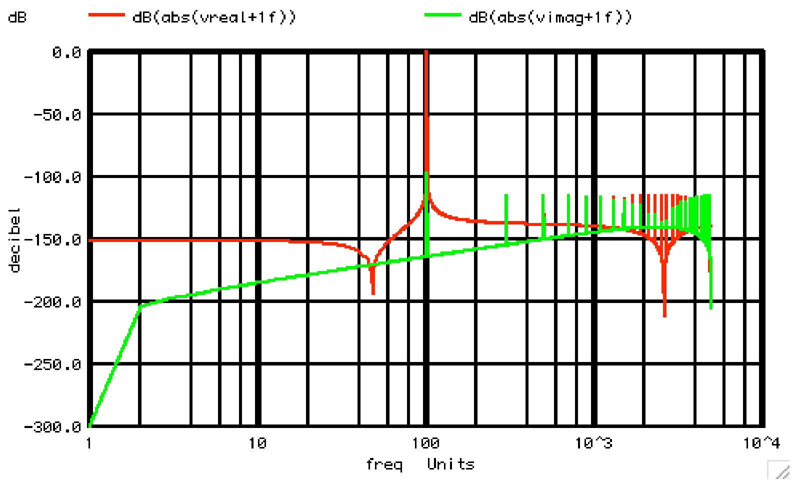
Distortion at 0.001% is about -100dB down. There are 10,000 data points which is 5,000 frequency bins. The flatband noise is around -140dB down and 5,000 frequency bin should raise it about 37dBs.

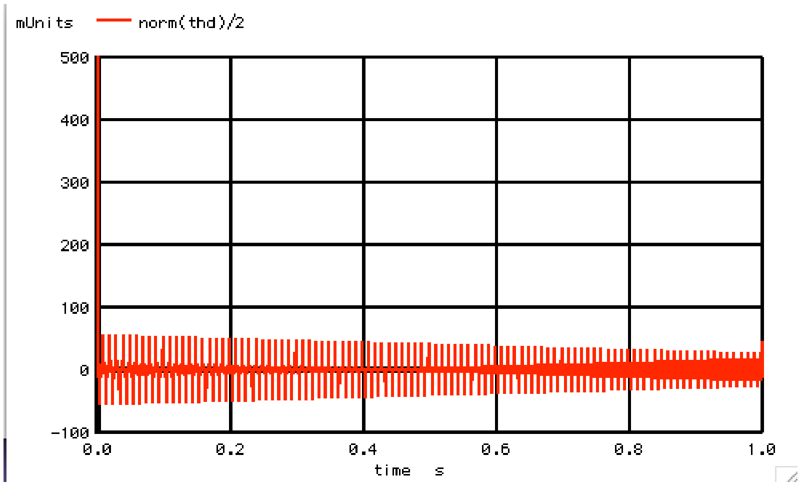


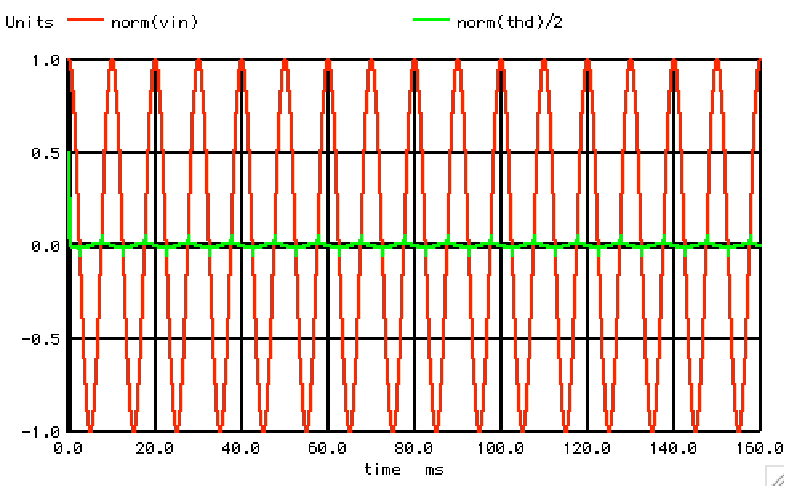
*=========When_does_the_THD_rise===================

The threshold appears to happen when the value of 2PI gets rolled off four digits to such..

```
BSIN      VIN    0      V =     cos(6.283185*v(VFreq)*v(Vtime) )
==============================================================
 THD_%=0.00143309  Fund_rms=0.70707  THD_rms=1.013E-05
```







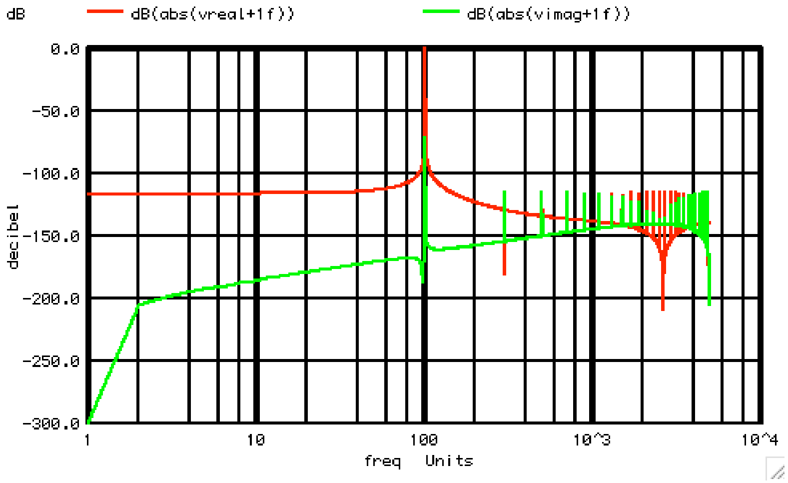
*=========Round_Off_Another_Digit===================

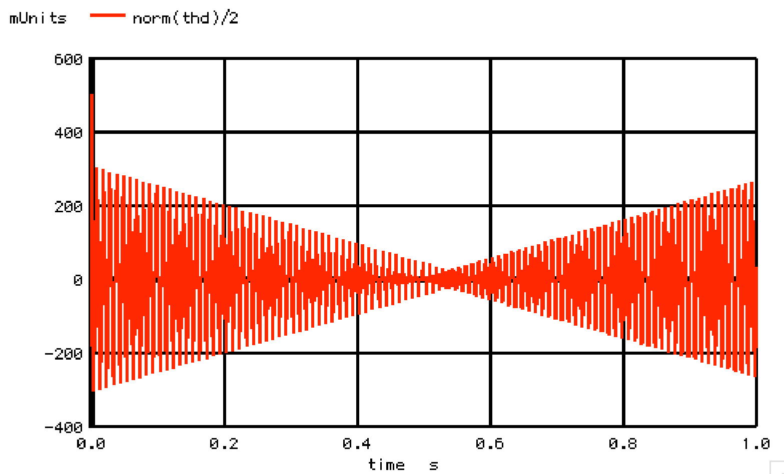
Another digit roll off, and another order of magnitude.

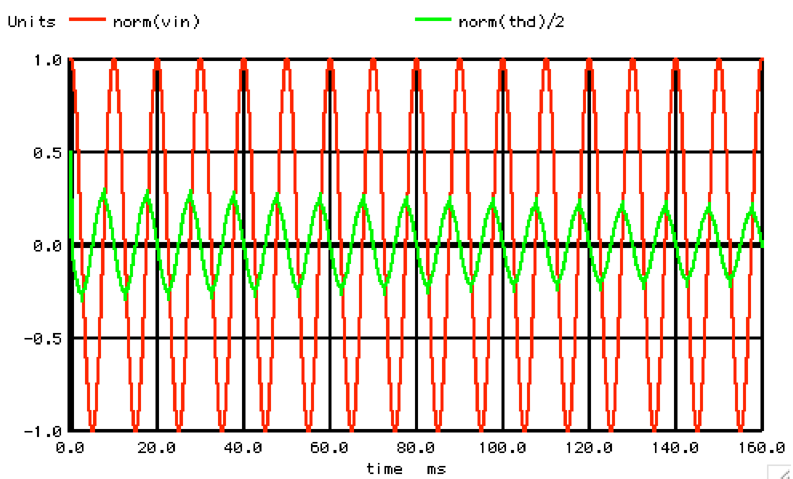
BSIN VIN 0 V = cos(6.28318*v(VFreq)*v(Vtime))

==

THD_%=0.0153607 Fund_rms=0.707074 THD_rms=0.000108612







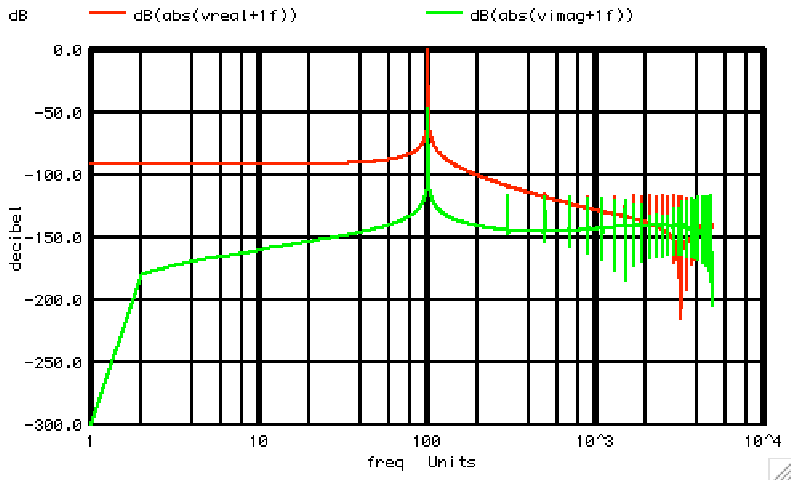
***==========Round_Off_Another_Digit====================**

And another digit and another order of magnitude.

BSIN VIN 0 V = cos(6.2831*v(VFreq)*v(Vtime))

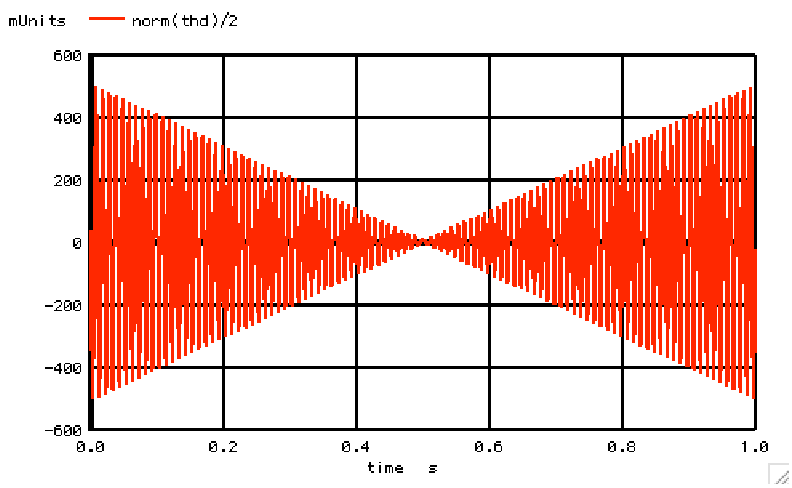
==

THD_%=0.246274 Fund_rms=0.707067 THD_rms=0.00174133

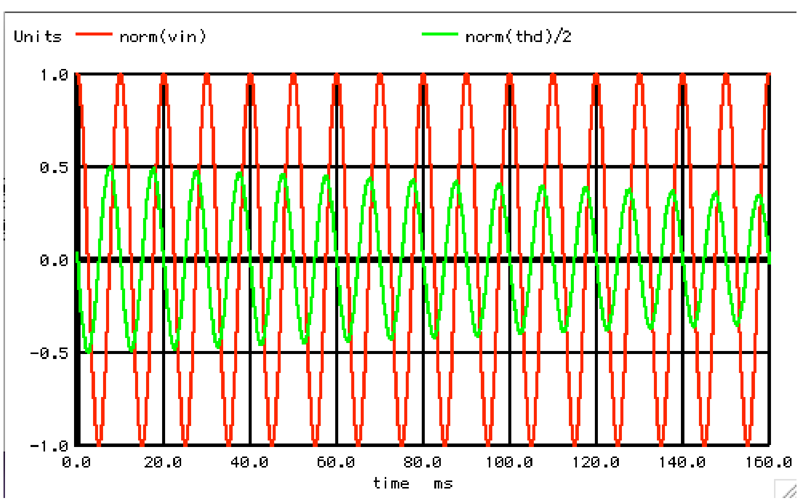


*=========Now_get_a_Better_Look_At_Leakage===================

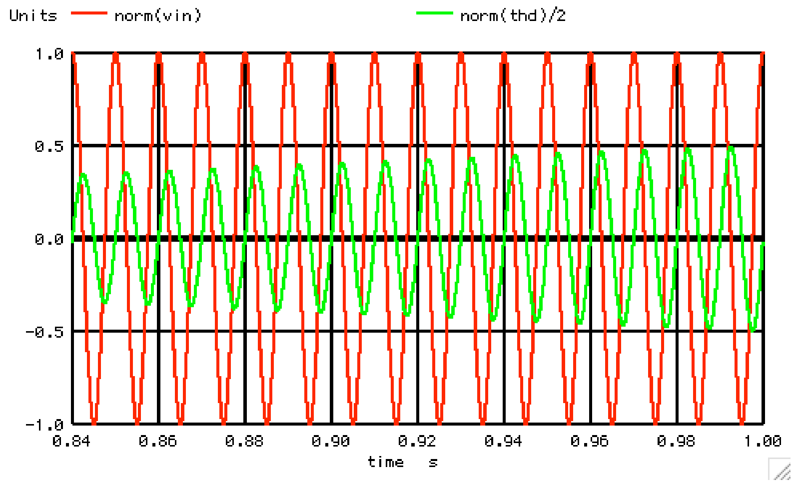
The spectrum leakage (THD) in time looks like a modulated signal.



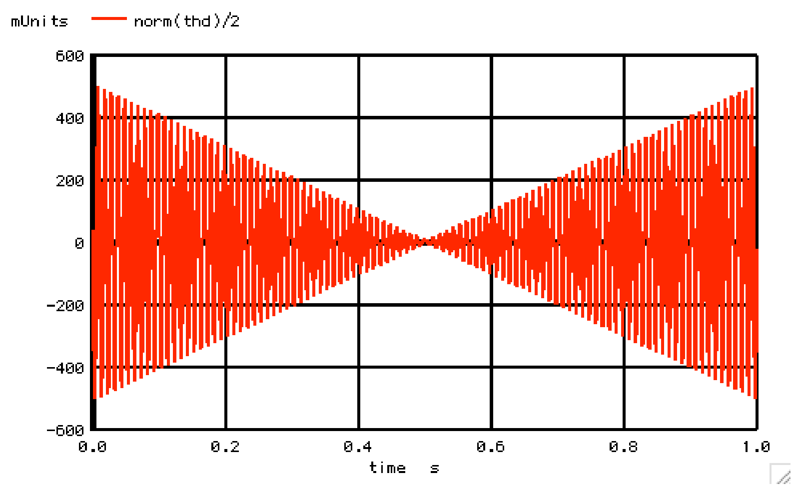
At time zero, the THD looks like fundamental with about a 90-degree phase lead.



At the end, the THD looks like fundamental with about a 90-degree phase lag.



So in the time domain, Spectrum leakage looks like fundamental with phase modulation and amplitude modulation.



===========Full_Netlist_For_Copy_Paste=======================

```
FFT_tests_SINE
VPI       VP      0      DC     3.141592653589793
VT        Vtime   0      DC     0       PWL ( 0 0 1 1 )
VFreq     VFreq   0      DC     100
*BSIN      VIN     0      V =    cos(6.2831853072*v(VFreq)*v(Vtime) )
BSIN      VIN     0      V =    cos(6.2831*v(VFreq)*v(Vtime) )
*6.283185307179586
.control
*TRAN     TSTEP  TSTOP  TSTART TMAX
tran     .1m     .9999  0      .1m
set       pensize = 2
linearize
plot      vin  xlimit 0 10m pointplot

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


let       sigAt                   = VFreq[0]
```

```
let       unvect                 = unitvec($&numb2)
let       fundspec               = unvect*0 +j(0)
let       fundspec[sigAt]        = real(ac_fft[sigAt])        +j(imag(ac_fft[sigAt] ))
let       fundspec[numb2-sigAt] = real(ac_fft[numb2-sigAt])  +j(imag(ac_fft[numb2-sigAt] ))
let       fund                   = ifft(fundspec)
let       dc_ofset               = real(ac_fft[0])
let       thdspec                = ac_fft
let       thdspec[0]             = 0        +j(0)
let       thdspec[sigAt]         = 0        +j(0)
let       thdspec[numb2-sigAt]  = 0        +j(0)
let       thd                    = ifft(thdspec)
plot      norm(vin) norm(thd)/2    xlimit 0 .15
plot      norm(vin) norm(thd)/2    xlimit .84 .9999
plot      norm(thd)/2

let       rms_Fund               = sqrt(mean(fund*fund))
let       rms_THD                = sqrt(mean(thd*thd))
let       THD_percent            = 100*rms_THD/rms_Fund
echo      " THD_%=$&THD_percent  Fund_rms=$&rms_Fund THD_rms=$&rms_THD   "

.endc
.end
```