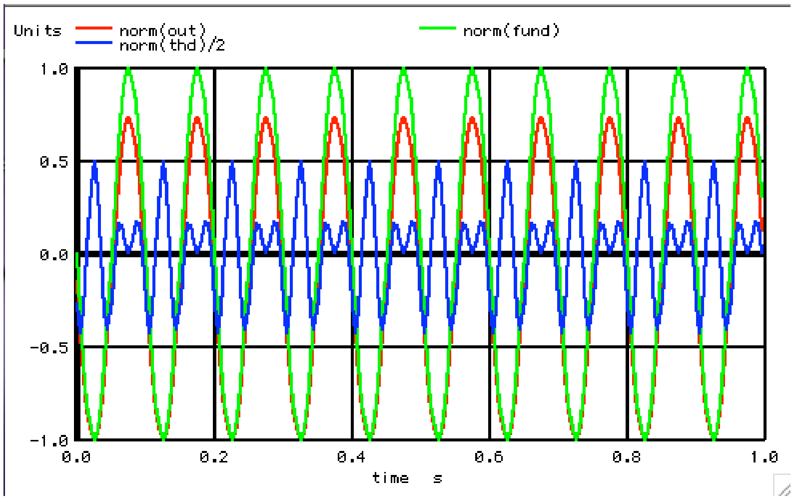
*=========**Input_Distortion_Waveforms**================

TWO WAY SPECTRUM TRANSFORMATION ALLOW HARMONICS TO BE VIEW AS WAVEFORMS SYNCHRONIZED TO THE SIGNAL GENERATING THEM. GREAT FOR TRACING DISTORTION BACK TO ITS SOURCE.



Simple input stages distort signal. And they can distort symmetrically (odd harmonics) or asymmetrically (even harmonics).

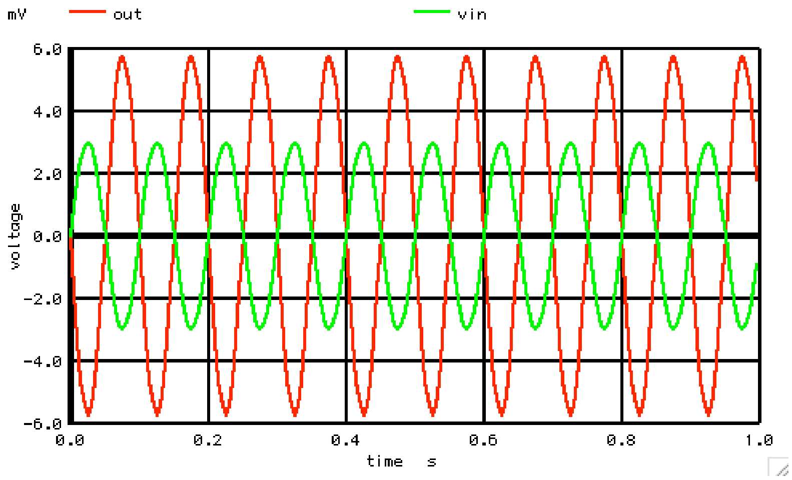
This simulation has been set up so that the DC voltages VGAIN and VOS control input levels and input offset.

==

```
INPUT_THD
*
*                VCC ____________________
*           R1      |                    |
*       _/\  /\  /\_|_/\  /\  /\_  R2    |
*       |  \/  \/         \/  \/  |      |
*       |          VC1            |    __|__
*       |___________       _______|   /VCC\
*    VB1            |     |    VC2    \___/
*     ________|' QN1  QN2`|_____       _|_
*    _|_      |`->    <-'|     _|_    ///
*   /VOS\        |______|     ///
*   \___/ VIN        _|_   VE
*    _|_            / _ \
*   /_  \ Vin       \/ \/
* //  \  \          /\_/\ ITAIL
* \    \//          \___/
*  \___/             _|_
*    _|_             ///
*    ///
*

*======== ====== ====== ====== ====== ====== ====== ====== ======
VCC       VCC    0      DC     10
VAC       VAC    0      DC     0      SIN(   0      1      10 )
VGAIN     VGAIN  0      DC     3m
VOS       VOS    0      DC     3u
BVIN      VIN    0      V      =  V(VGAIN)*V(VAC) + V(VOS)
QN1       VC1    VIN    VE     NPN1    1
QN2       VC2    0      VE     NPN1    1
IB1       VE     0      100u
R3        VCC    VC1    1K
R4        VCC    VC2    1K
BOUT      OUT    0      V = V(VC1)  -V(VC2)
.model    NPN1   NPN(   BF=210 VAF=216  )

.control
*TRAN     TSTEP  TSTOP  TSTART TMAX   ?UIC?
tran      5m     .995   0      5m
set       pensize = 2
linearize
plot      out   vin
```



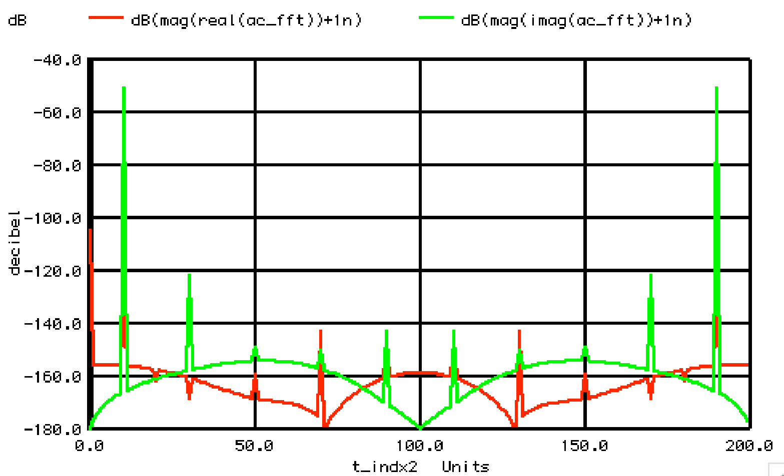
*=========The_distortion_levels_are_Low====================

Since the distortion levels are low, one can only view them on dB scaled spectrum outputs.

==

```
let       numb2 = length(vin)
print     numb2
let       t_indx2 = vector($&numb2)

let       ac = out +j(0)
let       ac_fft=fft(ac)
plot      dB(mag(real(ac_fft))+1n) dB(mag(imag(ac_fft))+1n) vs t_indx2
```



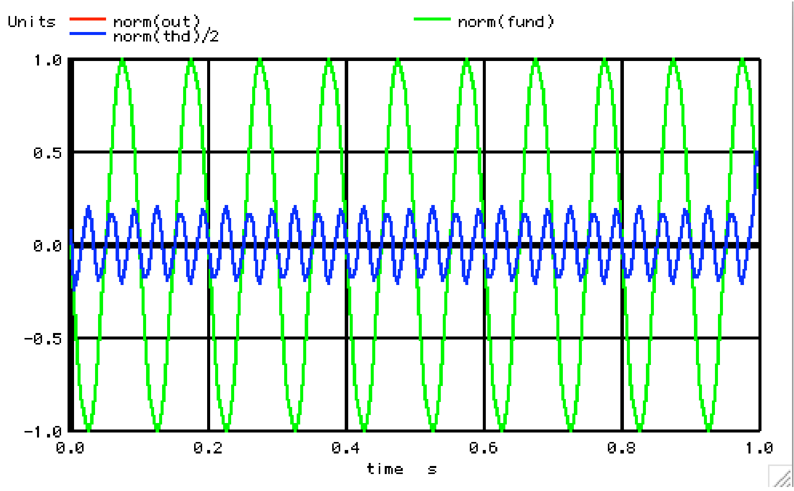
*=========Remove_Fundamental_to_see_THD====================

The IFFT function allows the fundamental to be removed so that the distortion waveform can be viewed.

==

```
let       sigAt                  =  10
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[10]             = 0         +j(0)
let       thdspec[numb2-10]       = 0         +j(0)
let       thd                     = ifft(thdspec)
```

```
plot      norm(out) norm(fund) norm(thd)/2
```



***=========Now_Calculate_THD===================**

And the vector processing functions make it easy to calculate distortion and output offset, etc..

```
==============================================================
let       rms_Fund                = sqrt(mean(fund*fund))
let       rms_THD                 = sqrt(mean(thd*thd))
let       THD_percent             = 100*rms_THD/rms_Fund
let       OFFS                    = VOS[0]
let       VINPK                   = VGAIN[0]
echo      "Offset=$&OFFS Vinpk=$&VINPK THD_%=$&THD_percent OutDC=$&dc_ofset"
==============================================================
```

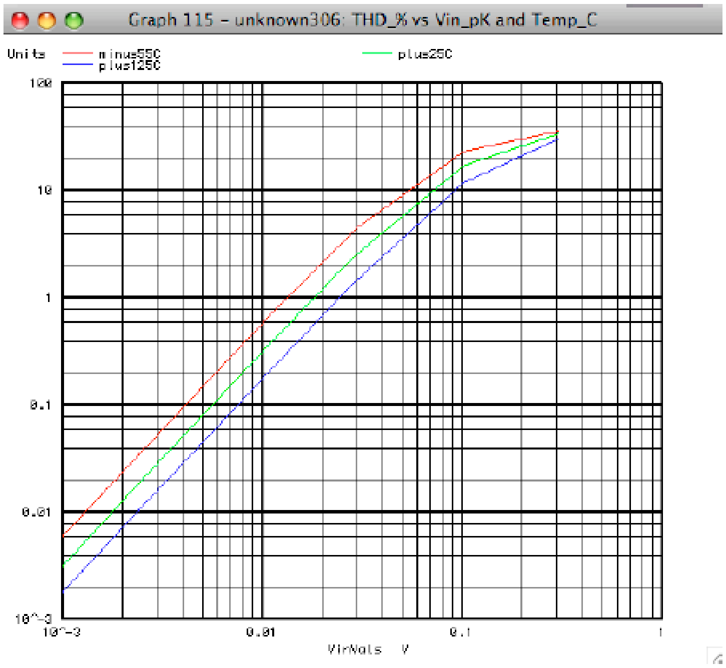
Offset=3E-06 Vinpk=0.003 THD_%=0.0285398 OutDC=-5.74604E-06

***=========Input_level_defines_THD===================**

The print out of the calculated values, together with the fact that the input offset and input signal magnitude are controlled by DC voltages VGAIN and VOS, make it easy to generate the following table.

```
==============================================================

Offset=3E-06 Vinpk=0.003 THD_%=0.0285398 OutDC=-5.74604E-06
Offset=3E-06 Vinpk=0.01  THD_%=0.296708  OutDC=-5.62098E-06
Offset=3E-06 Vinpk=0.018 THD_%=0.942284  OutDC=-5.36008E-06
Offset=3E-06 Vinpk=0.03  THD_%=2.55361   OutDC=-4.60556E-06
Offset=3E-06 Vinpk=0.1   THD_%=16.5428   OutDC=-6.41828E-07
```



The distortion versus input level can be sanity checked with the LM13600 datasheet, or with other spice simulations such as can be found here.

http://www.idea2ic.com/PlayWithSpice/pdf/DIFF_THD_TEMP.pdf

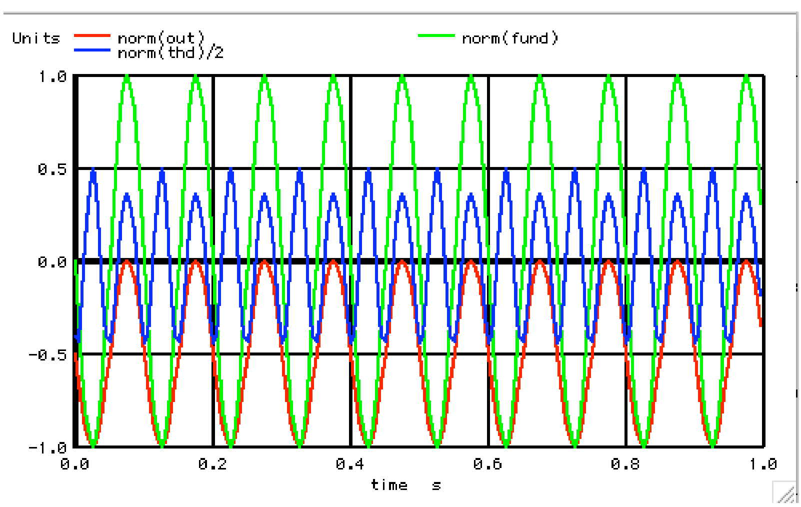
*=========Offset_levels_defines_THD_too====================

The reason viewing the distort is important is because things like input offset also create distortion, and of a different type.

In this case the input signals peak level and the input offset are both set to 3mV.

==

Offset=0.003 Vinpk=0.003 THD_%=0.167619 OutDC=-0.00575528



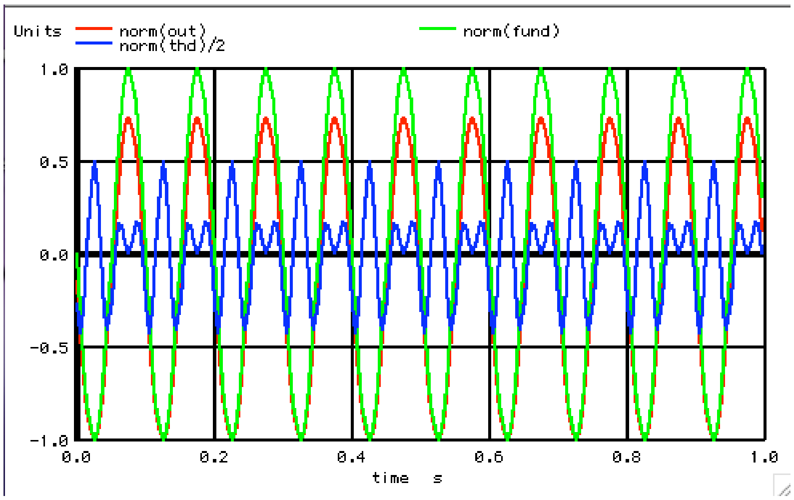
Distortion has just gotten 6 time worse, and viewing the THD shows that it is mainly second harmonic.

*=========When_signal_is_larger_than_offset====================

The input level starts to dominate the offset just about when it is 6 times larger than the offset. In this case the 18mV input level generates a distortion level which is 1.4 times the level without the input offset.

==

Offset=0.003 Vinpk=0.018 THD_%=1.33023 OutDC=-0.00544053



So being able to look at the distortion waveform can be very informative. Knowing whether the distortion is symmetrical or not gives a clue as to what is causing it.

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

```
Check_Timing_Linearity
*V_PULSE# NODE_P NODE_N DC     VALUE  PULSE( VINIT  VPULSE TDELAY TRISE  TFALL  PWIDTH PERIOD )
V_SQR     VIN    0      DC     0      PULSE( -1     1      -2.5m    1u     1u     5m     10m    )

.control
*TRAN     TSTEP  TSTOP  TSTART TMAX   ?UIC?
tran     .1m     1      0      .1m
set       pensize = 2
plot      vin  xlimit 0 10m pointplot

let       numb = length(vin)
print     numb
let       t_indx1 = vector($&numb)
let       t_ref   = vector($&numb)/$&numb
set       scale   t_indx1
plot      time -t_ref xlimit 0 20m


linearize
plot      vin  xlimit 0 10m pointplot
let       numb2 = length(vin)
print     numb2
let       numb3= numb2-1
let       t_indx2 = vector($&numb2)
let       t_ref2   = vector($&numb2)/$&numb3
set       scale t_indx2
plot      time -t_ref2


.endc
.end
```