*=========Output_Distortion================

The distortion that gets generated in an output stage can become important for things like audio power amplifiers. These amplifiers need rather large output transistors, and are connected in a AB bias fashion. Above 10KHz, there usually is a delay in how long it takes to turn a rather large transistor on. This cross over distortion is usually designed so that it is not visible to the eye. But it can be measured and viewed on a distortion analyzer.

This simulations emulates a lab distortion analyzer. The circuit is simple and simply puts the output transistors in a mode where they have a turn-on delay. Note the frequency is defined by a DC voltage source **Vfreq**.

==

```
Output_THD
*     ______________________
*    |                      |   _|_
*   _|_                     |  /VCC\
*  / _ \ IBIAS   QN1   QN2  |  \___/
*  \/ \/        ____        |   _|_
*  /\_/\     |    _|      _|    ///
*  \___/   VB1 |_|'______|'
*    |_________| |`->    |`->
*                   |       |  VOUT    ROUT
*     ________      |VAB    |____/\  /\  /\_
*    _|_      |     |       |      \/  \/   |
*  /_  \ VIN  |    <-     <-               _|_
* //  \  \    |_|'______|'   QP2           ///
* \    \//    |_|`_     |`_     ______
*  \___/      |____|     |   _|_    _|_
*   _|_                  |  /VEE\   ///
*   ///      QP!         |  \___/
*                        |____|
*


VCC       VCC     0      DC     10
VEE       VEE     0      DC     -10
VTime     VTime   0      DC     0      PWL(   0      0    1      1)
Vfreq     Vfreq   0      DC     8k
BVAC      IN      0      V  =   5*sin( 6.283185307179586*V(VFreq)*V(VTime))
QN1       VB1     VB1    VAB    NPN1   1
QN2       VCC     VB1    VOUT   NPN1   1
QP1       VIN     VIN    VAB    PNP1   1
QP2       VEE     VIN    VOUT   PNP1   1
IBIAS     VCC     VB1    900u
BOTA      VSS     0      I =    -3m*tanh((V(IN)-V(VOUT))*10)
RBP       VSS     VIN    5k
CBW       VIN     0      30p
Rout      VOUT    0      100
.model    NPN1    NPN(   BF=510 VAF=916 tf=100n  CJE=150p CJC=500p CJS=500p )
.model    PNP1    PNP(   BF=510 VAF=216 tf=1u    CJE=150p CJC=500p CJS=500p)
```

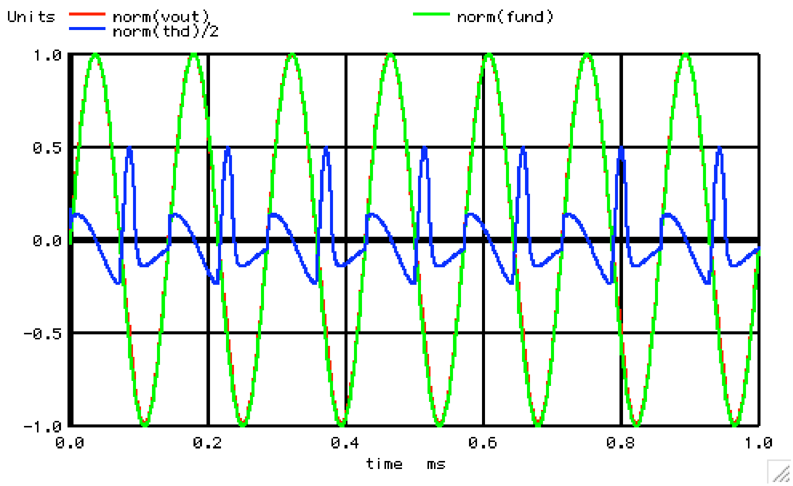
*=========Extract_the_Distortion====================

The FFT and IFFT functions allow the fundamental to be removed to view the distortion along side the output signal.

==

```
.control
*TRAN     TSTEP   TSTOP   TSTART TMAX    ?UIC?
tran      1u      .999m   0       1u
set       pensize = 2
linearize
let       numb2 = length(vin)
print     numb2
let       t_indx2 = vector($&numb2)
let       ac = vout +j(0)
let       ac_fft=fft(ac)
plot      real(ac_fft) imag(ac_fft) vs t_indx2
```

```
let       funBin                 = VFreq[0]/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_ofset               = 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(vout) norm(fund)  norm(thd)/2
```



*=========**Calculate_the_Distortion**====================

To put things into perspective, finding out what the actual distortion level is, determines what gets done about it.

==

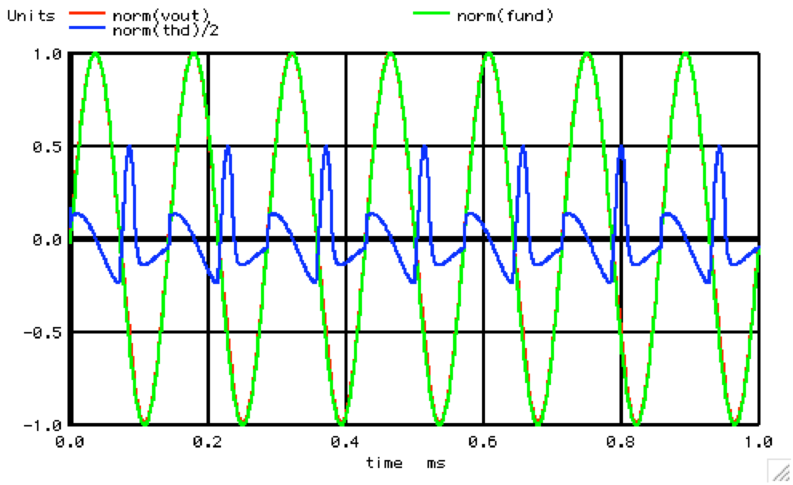
```
let       rms_Fund               = sqrt(mean(fund*fund))
let       rms_THD                = sqrt(mean(thd*thd))
let       THD_percent            = 100*rms_THD/rms_Fund
let       FREQ_Hz                = VFreq[0]
echo      "Freq_Hz=$&FREQ_Hz     THD_percent=$&THD_percent    DC=$&dc_ofset"
```

==

Freq_Hz=8000 THD_percent=4.2893   DC=0.105696

*=========**What_does_4%_distortion_mean**====================

If one is young, one might be able to hear the distortion of a 8KHz signal. But usually the distortion gets reduced to make pretty distortion plot.

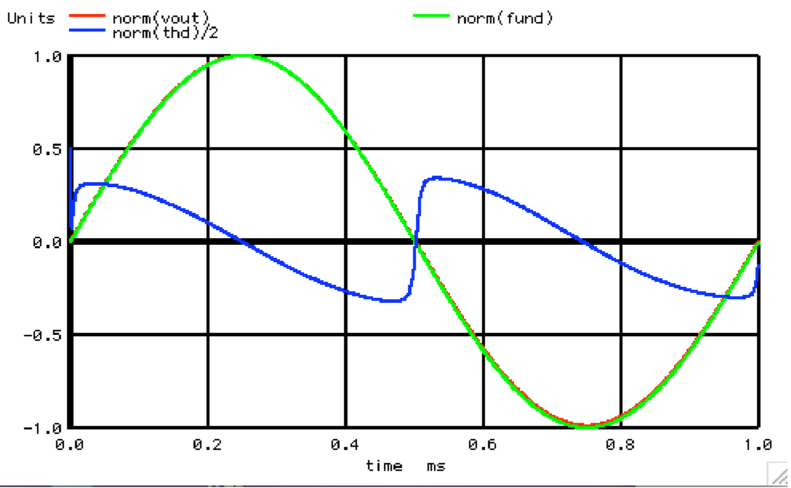


*==========Distortion_is_frequency_dependent====================

At 1kHz, the distortion is 42 times lower than at 8Khz. Different parts of an audio power amplifier require attention to distortion at different frequencies.

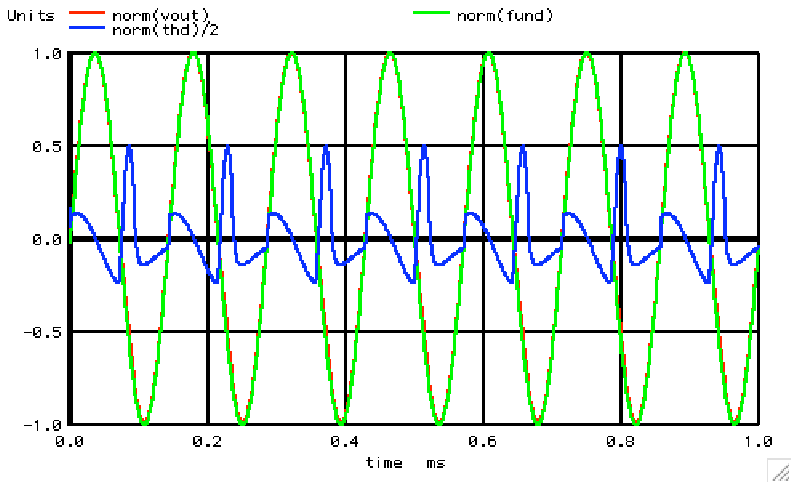
==

Freq_Hz=1000 THD_percent=0.101815 DC=0.0310647



*==========The_distortion_waveform====================

The distortion wave form shows where a transistor is working hard. For audio power amplifiers like the LM383, there is a tradeoff in output distortion versus supply current. Running an output more B bias will increase distortion, but lower supply current. While power amplifiers can put out several amps, having the lowest supply current as possible can sell the amplifier.



The distortion waveform provides the ability to do cause and effect analysis on a circuit design. Making it possible to see when the distortion happens, it is not real hard to find something going on inside a circuit design which is happening at the same time. Not having to guess is a real time saver.

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

```
Output_THD
*     __________________________
*    |                     |   _|_
*   _|_                    |  /VCC\
*  / _ \ IBIAS   QN1   QN2 |  \___/
*  \/ \/        ____       |   _|_
*  /\_/\       |   _|     _|   ///
*  \___/   VB1 |_|'_____|'
*    |_________| |`->   |`->
*                   |      | VOUT   ROUT
*     ________      |VAB   |____/\  /\  /\_
*   _|_       |     |      |      \/  \/   |
*  / _ \ VIN  |    <-    <-               _|_
* // \  \     |_|'_____|'   QP2           ///
* \   \//     | |`_    |`_      _____
*  \___/      |____|      |   _|_   _|_
*   _|_                   |  /VEE\  ///
*   ///       QP!         |  \___/
*                         |____|
*
*


VCC       VCC    0      DC     10
VEE       VEE    0      DC     -10
VTime     VTime  0      DC     0      PWL(   0      0    1      1)
Vfreq     Vfreq  0      DC     1k
BVAC      IN     0      V  =   5*sin( 6.283185307179586*V(VFreq)*V(VTime))
QN1       VB1    VB1    VAB    NPN1   1
QN2       VCC    VB1    VOUT   NPN1   1
QP1       VIN    VIN    VAB    PNP1   1
QP2       VEE    VIN    VOUT   PNP1   1
IBIAS     VCC    VB1    900u
BOTA      VSS    0      I =    -3m*tanh((V(IN)-V(VOUT))*10)
RBP       VSS    VIN    5k
CBW       VIN    0      30p
Rout      VOUT   0      100
.model    NPN1   NPN(   BF=510 VAF=916 tf=100n  CJE=150p CJC=500p CJS=500p )
.model    PNP1   PNP(   BF=510 VAF=216 tf=1u    CJE=150p CJC=500p CJS=500p)

.control
*TRAN     TSTEP  TSTOP  TSTART TMAX   ?UIC?
tran      1u     .999m  0       1u
set       pensize = 2
linearize
let       numb2 = length(vin)
print     numb2
let       t_indx2 = vector($&numb2)
let       ac = vout +j(0)
```

```
let      ac_fft=fft(ac)
plot     real(ac_fft) imag(ac_fft) vs t_indx2
let      funBin                 = VFreq[0]/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_ofset               = 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(vout) norm(fund)  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
let      FREQ_Hz                = VFreq[0]
echo     "Freq_Hz=$&FREQ_Hz THD_percent=$&THD_percent   DC=$&dc_ofset"

.endc
.end
```