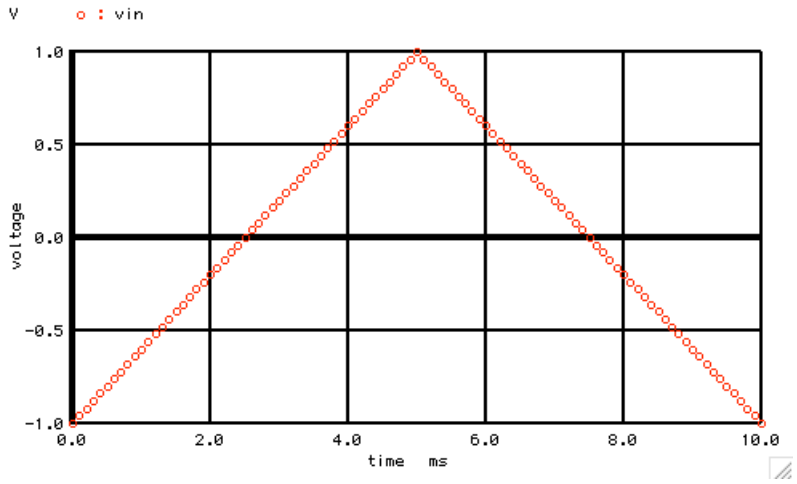
*=========FFT_TRI================

There are ways to further automate the use of the fft/ifft functions. In this case a DC voltage Vfreq is controlling the frequency.

*==========Create_Triangle_WaveForm====================

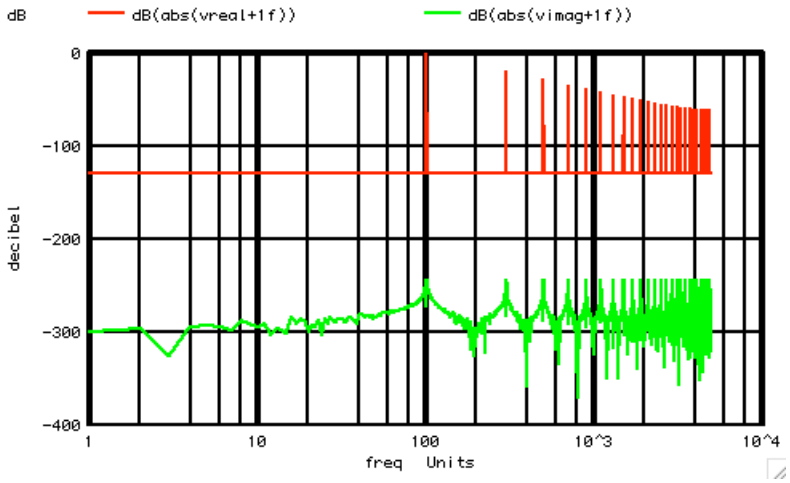
```
VPI        VP      0       DC      3.141592653589793
VT         Vtime   0       DC      0       PWL ( 0 0 1 1 )
VFreq      VFreq   0       DC      100
BTRI       VIN     0       V =     2*acos(cos(6.283185307179586*v(VFreq)*v(Vtime)))/v(VP) - 1

.control
*TRAN      TSTEP   TSTOP   TSTART  TMAX
tran      .1m      .9999   0       .1m
set        pensize = 2
linearize
plot       vin   xlimit 0 10m pointplot
```



*==========Translate_into_dB_freq====================

This spectrum output appears to have about a 130dB range.

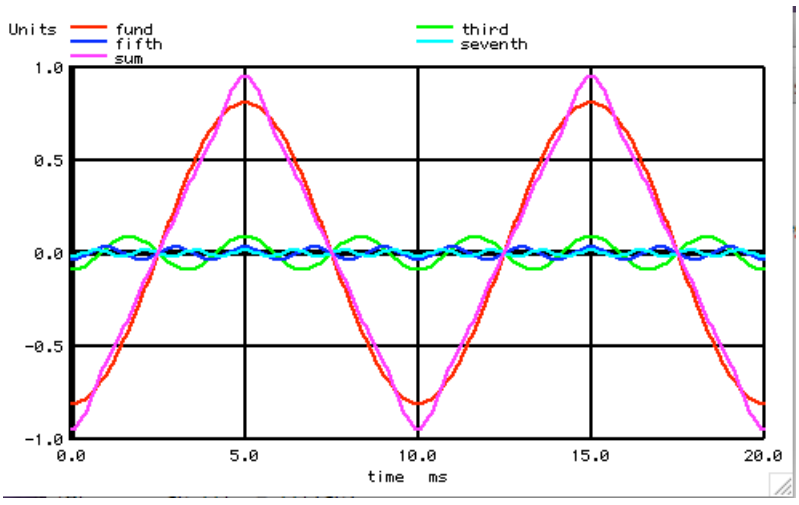
```
=================================================================
let        numb2     = length(vin)
print      numb2
let        ac        = vin +j(0)
let        ac_fft    = fft(ac)
let        numb_f    = (numb2)/2
compose    freq      start = 1 stop = $&numb_f  step =1
compose    vreal     start = 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
```



*=========Now_Dissect_the_Harmonics====================

Now the dissection of the harmonics is even more automated.

```
=================================================================
let       funb                        = Vfreq[0]
let       unvect                      = unitvec($&numb2)
let       fundspec                    = unvect*0 +j(0)
let       fundspec[funb]              = real(ac_fft[funb])         +j(imag(ac_fft[funb] ))
let       fundspec[numb2-funb]        = real(ac_fft[numb2-funb])   +j(imag(ac_fft[numb2-funb] ))
let       fund =  ifft(fundspec)
let       thirdspec                   = unvect*0 +j(0)
let       thirdspec[3*funb]           = real(ac_fft[3*funb])         +j(imag(ac_fft[3*funb] ))
let       thirdspec[numb2-3*funb]     = real(ac_fft[numb2-3*funb])   +j(imag(ac_fft[numb2-3*funb] ))
let       third                       = ifft(thirdspec)
let       fifthspec                   = unvect*0 +j(0)
let       fifthspec[5*funb]           = real(ac_fft[5*funb])         +j(imag(ac_fft[5*funb] ))
let       fifthspec[numb2-5*funb]     = real(ac_fft[numb2-5*funb])   +j(imag(ac_fft[numb2-5*funb] ))
let       fifth                       = ifft(fifthspec)
let       seventhspec                 = unvect*0 +j(0)
let       seventhspec[7*funb]         = real(ac_fft[7*funb])         +j(imag(ac_fft[7*funb] ))
let       seventhspec[numb2-7*funb]   = real(ac_fft[numb2-7*funb])   +j(imag(ac_fft[numb2-7*funb] ))
let       seventh                     = ifft(seventhspec)
let       sum  = fund + third + fifth + seventh
set       scale  time
plot      fund  third fifth seventh  sum xlimit 0 20m
```



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

```
FFT_tests_TRI
VPI       VP     0      DC     3.141592653589793
VT        Vtime  0      DC     0       PWL ( 0 0 1 1 )
```

```
VFreq     VFreq  0      DC     100
BTRI      VIN    0      V =    2*acos(cos(6.283185307179586*v(VFreq)*v(Vtime)))/v(VP) - 1

.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       funb                       = Vfreq[0]
let       unvect                     = unitvec($&numb2)
let       fundspec                   = unvect*0 +j(0)
let       fundspec[funb]             = real(ac_fft[funb])        +j(imag(ac_fft[funb] ))
let       fundspec[numb2-funb]       = real(ac_fft[numb2-funb])  +j(imag(ac_fft[numb2-funb] ))
let       fund =  ifft(fundspec)
let       thirdspec                  = unvect*0 +j(0)
let       thirdspec[3*funb]          = real(ac_fft[3*funb])        +j(imag(ac_fft[3*funb] ))
let       thirdspec[numb2-3*funb]    = real(ac_fft[numb2-3*funb])  +j(imag(ac_fft[numb2-3*funb] ))
let       third                      = ifft(thirdspec)
let       fifthspec                  = unvect*0 +j(0)
let       fifthspec[5*funb]          = real(ac_fft[5*funb])        +j(imag(ac_fft[5*funb] ))
let       fifthspec[numb2-5*funb]    = real(ac_fft[numb2-5*funb])  +j(imag(ac_fft[numb2-5*funb] ))
let       fifth                      = ifft(fifthspec)
let       seventhspec                = unvect*0 +j(0)
let       seventhspec[7*funb]        = real(ac_fft[7*funb])        +j(imag(ac_fft[7*funb] ))
let       seventhspec[numb2-7*funb]  = real(ac_fft[numb2-7*funb])  +j(imag(ac_fft[numb2-7*funb] ))
let       seventh                    = ifft(seventhspec)
let       sum  = fund + third + fifth + seventh
set       scale  time
plot      fund  third fifth seventh  sum xlimit 0 20m

.endc
.end
```