===========**MacSpice_Spectrum_Scale**=======================

Using the EULER_Format in spice may create a lot of confusion. Unless it is done in 3D, one tends to have to keep a lot of things in one's head to use it right.

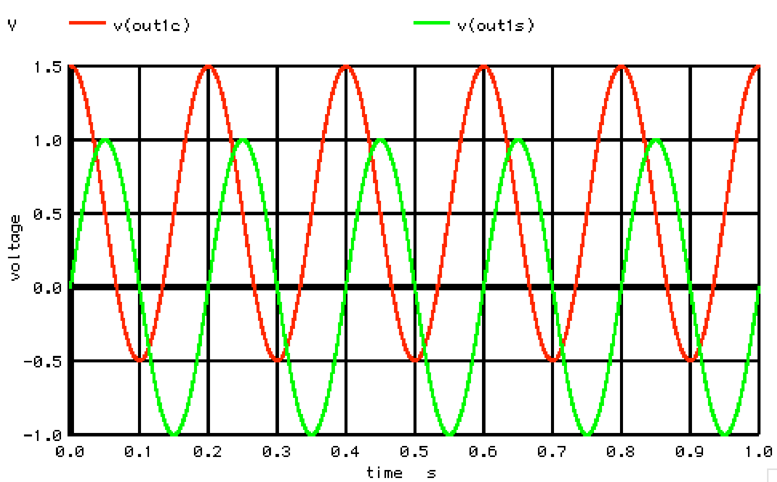
This is how MacSpice formats things.

```
FT_Scale_tests
* dsauersanjose@aol.com  1/21/10
* www.idea2ic.com
*                          the spectrum output appears to have..
*                          1Vpk cosine => real_mag 1 => real_rms .707
*     Vtime        ^  OUT  1Vpk sine   => imag_mag 1 => imag_rms .707
*                 / \
*   VT   _|_   |  _|_      frequency(0) = out of range
*       /   \  | /   \     (No DC term )
*      //\   \ |_//\  \
*      \   \//   \   \//
*       \___/     \___/ B1
*         |         |
*         |         |
*        ///       ///
*
.Option   srcsteps = 1     set Gmin = 1.0000E-02
*=========Circuit_Netlist=========================
VT        Vtime      0     PWL     ( 0  0 1  1 )
B1C       OUT1C      0     V =     cos(2*3.14159*5*v(Vtime)) +.5
B1S       OUT1S      0     V =     sin(2*3.14159*5*v(Vtime))
*=================================================
```

Creating waveforms in this format can make it easier to see everything.

```
*=================================================================
*TRAN     TSTEP       TSTOP  TSTART  TMAX     ?UIC?
.tran     0.1m        1      0       0.1m
.control
run
set       pensize =   2
plot      v(out1c)    v(out1s)
*=================================================================
```

Set total time to 1 second. Bin resolution = 1Hz
Set sample rate to .01m   Nyquist =   5KHz



===========**RMS_the_Input_SineWave**=======================

One can RMS the input Array OUT1S[] to get the expected .707 value out for a unity magnitude.

```
*===========Find_RMS_OUT1S========================
let num =   length(out1s)-1
```

```
let i  =    0
let vpwr =  0
repeat      $&num
let i =     i +1
let vpwr =  vpwr  + (mag(OUT1S[i])*mag(OUT1S[i]))/num
end
let vrms1 = sqrt(vpwr)
echo        INPUT RMS =  $&vrms1
*=============================================================
```

INPUT RMS = 0.706863

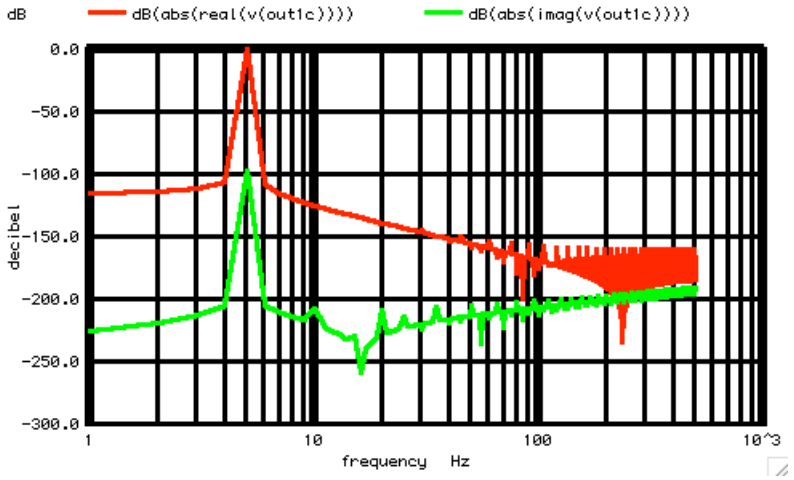
==========Spectrum_and_Plot_The_Cosine==========

Set total time to 1 second. Bin resolution = 1Hz
Set sample rate to .1m   Nyquist =   5KHz

Setting Spectrum max frequency to 500

```
*=========Find_Cosine_Spectrum=========================
linearize
*SPEC     FSTART     FSTOP FSTEP  VECTOR
spec      1          500   1      v(out1c)
set       pensize =  3
plot      dB(abs(real(v(out1c)))) dB(abs(imag(v(out1c))))   xlog
*=============================================================
```

Log or dB scale plots cannot plot anything less than zero.
So negative values need to be absolute valued.



==========Print_Out_Some_Spectrum_Values==========

The spectrum can to be printed out to show all the values.

```
*=========Print_Cosine_Spectrum=========================
foreach   ii 1 2 3 4 5 6 7
let i =   $ii -1
let fr=   frequency[i]
let       vfftr = real(v(out1c)[i])
let       vffti = imag(v(out1c)[i])
echo      index= $&i   freq= $&fr   real=  $&vfftr  imag=  $&vffti
end
*=============================================================
```

But one can get at all important details by using
the foreach on the frequency array and the output array.
The following is the MacSpice Printout.

```
index = 0 freq = 1 real = -1.76235E-06 imag = -4.67344E-12
index = 1 freq = 2 real = -2.01374E-06 imag = -1.06821E-11
index = 2 freq = 3 real = -2.64221E-06 imag = -2.10302E-11
```

```
index = 3 freq = 4 real = -4.69524E-06 imag = -4.98467E-11
index = 4 freq = 5 real = 1            imag = 1.32779E-05
index = 5 freq = 6 real = 3.83675E-06  imag = 6.11724E-11
index = 6 freq = 7 real = 1.75709E-06  imag = 3.27082E-11
```

Notice index zero starts at 1Hz.
The total time of 1sec sets the FFT Bin_Width to 1Hz.

A foreach can step through particular frequencies.

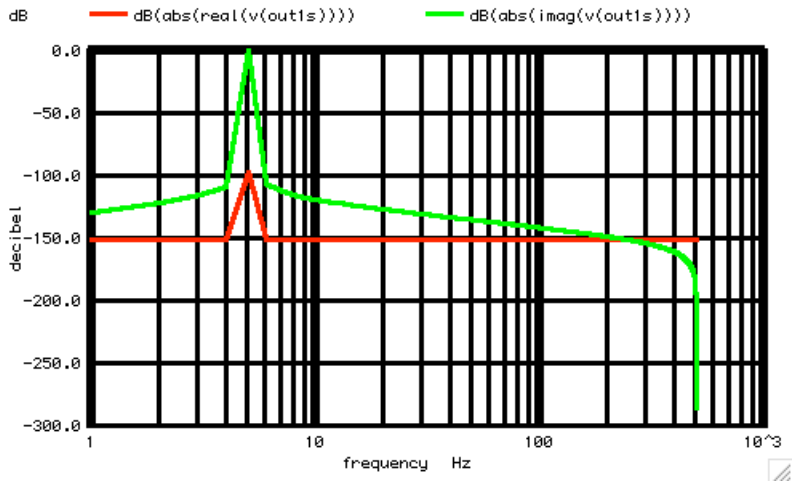
A unity gain 5Hz Cosine come out as one at freq = 5.

Note there is no DC (frequency=0) term.
This avoids some loglog plotting problems.

==========Now_Spectrum_and_Plot_The_Sine==========

Need to rerun to fill array out1s.

```
*=========ReRun_Simulation=========================
run
*=========Find_Sine_Spectrum=======================
linearize
set       specwindow=   "none"
spec      1          500   1     v(out1s)
set       pensize =  3
plot      dB(abs(real(v(out1s)))) dB(abs(imag(v(out1s))))   xlog
*=================================================================
```



==========Print_Out_Some_Spectrum_Values==========

Because the dynamic range of the signal is so high,
the dB scale is used and the real or imaginary values
can go negative. The absolute value function can avoid
plot problems.

```
*=========Print_Sine_Spectrum======================
foreach   ii 1 2 3 4 5 6 7
let i =   $ii -1
let fr=   frequency[i]
let       vfftr = real(v(out1s)[i])
let       vffti = imag(v(out1s)[i])
echo      index= $&i   freq= $&fr   real=  $&vfftr  imag=  $&vffti
end
*=================================================================
```

A unity gain 5Hz Sine comes out as imaginary one at freq = 5.
Note there is no DC (frequency=0) term.

The following is the MacSpice Printout.

```
index = 0 freq = 1 real = -2.65005E-08 imag = -3.51973E-07
index = 1 freq = 2 real = -2.64972E-08 imag = -8.04502E-07
index = 2 freq = 3 real = -2.64888E-08 imag = -1.58384E-06
index = 3 freq = 4 real = -2.64616E-08 imag = -3.75421E-06
index = 4 freq = 5 real = -1.33044E-05 imag = 1
index = 5 freq = 6 real = -2.65749E-08 imag = 4.60709E-06
index = 6 freq = 7 real = -2.65473E-08 imag = 2.46341E-06
```

```
===========Full_Netlist_For_Copy_Paste========================
FFT_Scale_tests
* dsauersanjose@aol.com  1/21/10
* www.idea2ic.com
*                         the spectrum output appears to have..
*                         1Vpk cosine => real 1
*     Vtime       ^  OUT  1Vpk sine   => imag 1
*         ____    / \
*  VT  _|_    |   _|_      frequency(0) = out of range
*     /_  \   |  /_  \     (No DC term )
*    // \ \  |_// \ \
*    \   \//   \   \//
*     \___/     \___/ B1
*       |         |
*       |         |
*      _|_       _|_
*      ///       ///
*
.Option   srcsteps = 1     set Gmin = 1.0000E-02
*=========Circuit_Netlist=========================
VT        Vtime      0     PWL    ( 0 0 1 1 )
B1C       OUT1C      0     V =    cos(2*3.14159*5*v(Vtime)) +.5
B1S       OUT1S      0     V =    sin(2*3.14159*5*v(Vtime))
B2        OUT        0     V =    v(OUT1C) + v(OUT1C)
.tran     1m         1     0     .1m
.control
run
set       pensize =  2
plot      v(out1c)   v(out1s)
*=========Find_Cosine_Spectrum=========================
linearize
set       specwindow=   "none"
spec      1          500   1     v(out1c)
set       pensize =  3
plot      dB(abs(real(v(out1c)))) dB(abs(imag(v(out1c))))   xlog
*=========Print_Cosine_Spectrum=========================
foreach   ii 1 2 3 4 5 6 7
let i =   $ii -1
let fr=   frequency[i]
let       vfftr = real(v(out1c)[i])
let       vffti = imag(v(out1c)[i])
echo      index= $&i   freq= $&fr   real=  $&vfftr  imag=  $&vffti
end
*=========ReRun_Simulation=========================
run
*=========Find_Sine_Spectrum=========================
linearize
set       specwindow=   "none"
spec      1          500   1     v(out1s)
set       pensize =  3
plot      dB(abs(real(v(out1s)))) dB(abs(imag(v(out1s))))   xlog
*=========Print_Sine_Spectrum=========================
foreach   ii 1 2 3 4 5 6 7
let i =   $ii -1
let fr=   frequency[i]
let       vfftr = real(v(out1s)[i])
let       vffti = imag(v(out1s)[i])
echo      index= $&i   freq= $&fr   real=  $&vfftr  imag=  $&vffti
end
.endc
.end
```

2.12.10_2.33PM

dsauersanjose@aol.com
Don Sauer
http://www.idea2ic.com/