

***=====Dual_Purpose_Windowing=====**

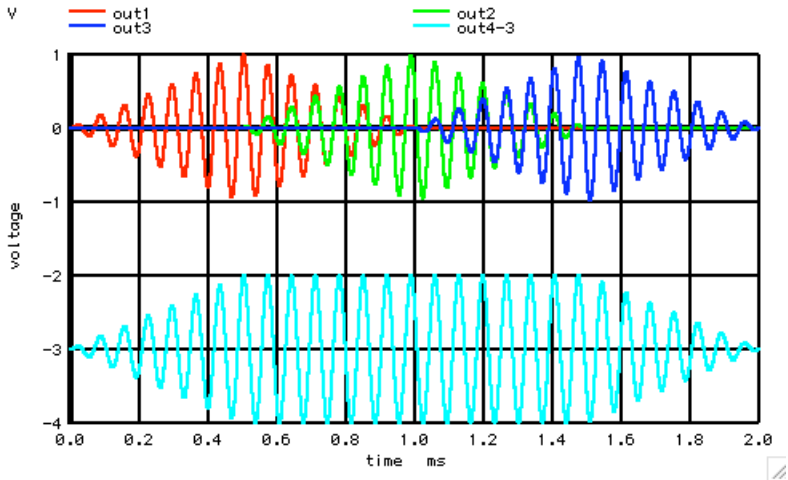
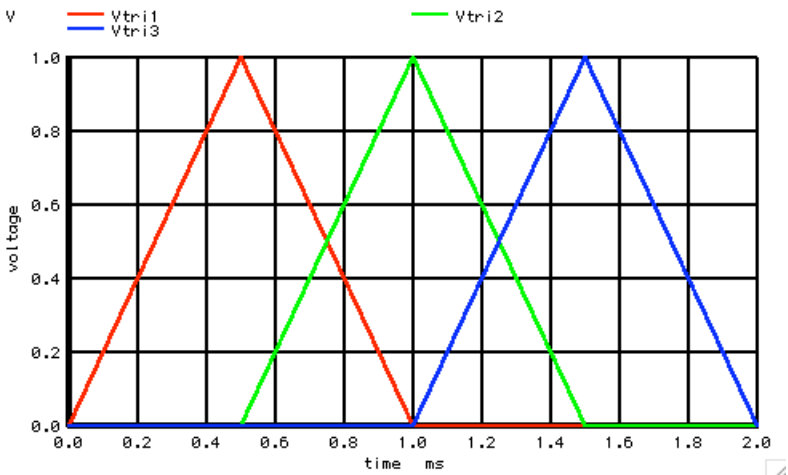
The FFT transform has the disadvantage that most real signals have changing spectrums over time. Spectrums of voice or music really need to be looked over short periods of time because much of the information is encoded as when the spectrum changes.

One convenient way to create small windows of time is to ramp up and ramp down a signal's amplitude.

```

=====
*V SIN#  NODE_P  NODE_N  DC    VALUE  SIN(  V_DC  AC_MAG  FREQ  DELAY  FDamp)
Vsig    Vsig     0      DC    0      SIN(  0     1     14.4k  )
Vtri1   Vtri1    0      DC    0      PWL(  0     0     0.5m   0     .5m  1    1m  0  )
Vtri2   Vtri2    0      DC    0      PWL(  0     0     1.0m   0     1m   1    1.5m 0  )
Vtri3   Vtri3    0      DC    0      PWL(  0     0     1.0m   0     1.5m 1    2.0m 0  )
Btri1   OUT1     0      V =   V(Vsig)*V(Vtri1)
Btri2   OUT2     0      V =   V(Vsig)*V(Vtri2)
Btri3   OUT3     0      V =   V(Vsig)*V(Vtri3)
Btri4   OUT4     0      V =   V(OUT1) +V(OUT2) +V(OUT3)
=====

```

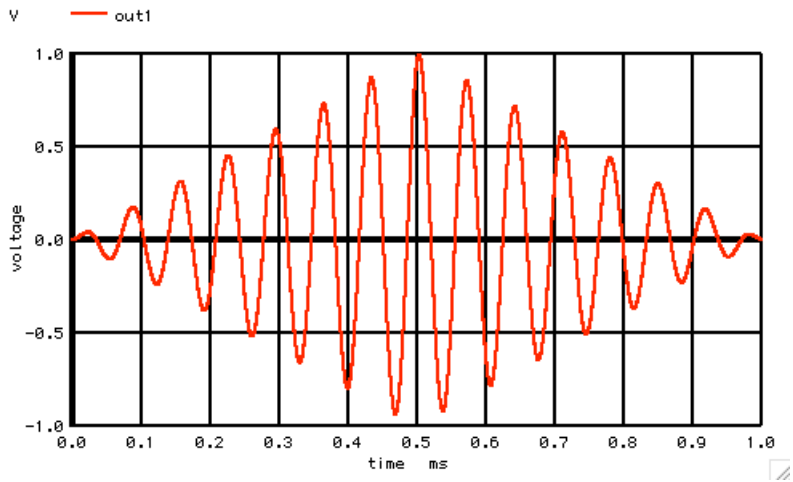
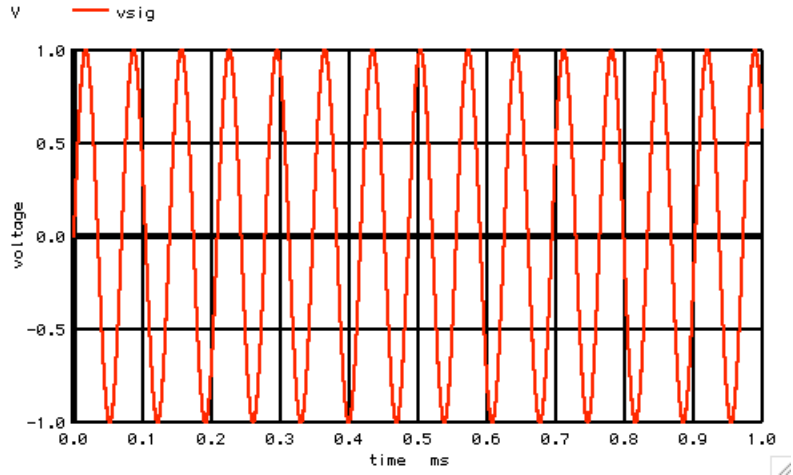


Overlapping the ramp up and down windows allows a signal to be split into separate time periods which can easily be put back together.

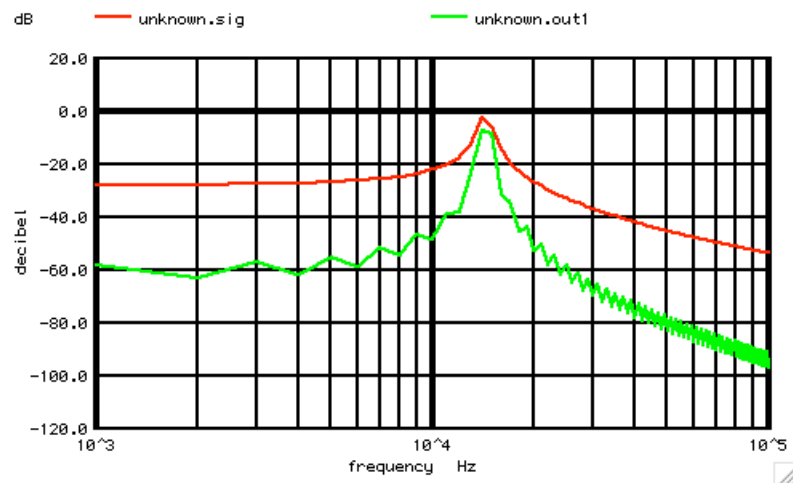
There is another problem if the signal does not contain a whole number of cycles in the sample period. This causes the spectrum output to leak across the whole spectrum.



*V_SIN#	NODE_P	NODE_N	DC	VALUE	SIN(V_DC	AC_MAG	FREQ	DELAY	FDamp)
Vsig	Vsig	0	DC	0	SIN(0	1	14.4k)



The process of ramping up and down the input signal also tends to reduce this Spectrum Leakage as well.



The triangle window is good for overlapping simplicity.
 And it also greatly improves the Spectrum leakage issues.
 The Hanning window, which uses more of a cosine window,
 can show in better detail what is going on.

=====**Full_Netlist_For_Copy_Paste**=====

```
Dual_Purpose_Windowing
.Option srcsteps = 1 set Gmin = 1.0000E-02
*=====Circuit_Netlist=====
*V SIN#  NODE_P  NODE_N  DC    VALUE  SIN(  V_DC  AC_MAG  FREQ  DELAY  FDamp)
Vsig    vsig    0      DC    0      SIN(  0      1      14.4k  )
Vtri1   Vtri1   0      DC    0      PWL(  0      0      0.5m   0      1m   1   1.5m  0  )
Vtri2   Vtri2   0      DC    0      PWL(  0      0      1.0m   0      1.5m 1   2.0m  0  )
Vtri3   Vtri3   0      DC    0      PWL(  0      0      1.0m   0      1.5m 1   2.0m  0  )
Btri1   OUT1    0      V =   V(Vsig)*V(Vtri1)
Btri2   OUT2    0      V =   V(Vsig)*V(Vtri2)
Btri3   OUT3    0      V =   V(Vsig)*V(Vtri3)
Btri4   OUT4    0      V =   V(OUT1) +V(OUT2) +V(OUT3)

.control
*TRAN  TSTEP  TSTOP  TSTART TMAX ?UIC?

tran   .1u    2m    0      .1u
set    pensize = 2
plot   Vtri1 Vtri2 Vtri3
plot   out1 out2 out3 out4-3

setplot new
let    "sig"  =    0*vector(100)
let    "out1" =    0*vector(100)

tran   .1u    1m    0      .1u
plot   vsig
set    specwindow= "rectangular"
spec   1k      100k  1k      v(vsig)
let    unknown.sig =    dB(mag(v(vsig)))

tran   .1u    1m    0      .1u
plot   out1
set    specwindow= "rectangular"
spec   1k      100k  1k      v(out1)
let    unknown.out1 =    dB(mag(v(out1)))

plot   unknown.sig  unknown.out1  vs frequency xlog ylimit -120 20

.endc
.end

7.29.10_12.02PM
dsauersanjose@aol.com
Don Sauer
http://www.idea2ic.com/
```