```
*============Transient_Timing_SineWave_1msec===========
```

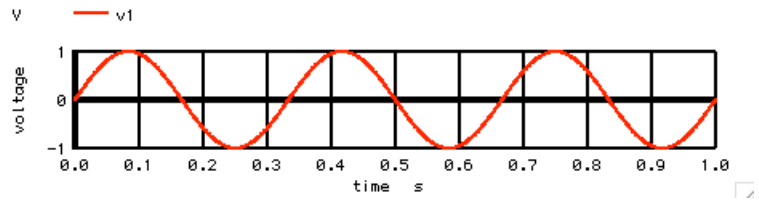
Sine-waves at higher tmax values make it harder to see any timing inconsistencies.

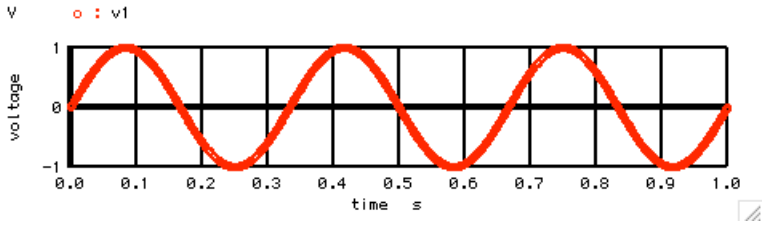
```
=========================================================================
*V_SIN#    NODE_P NODE_N DC      VALUE   SIN(    V_DC    AC_MAG FREQ    DELAY   FDamp)
V_SIN      V1     0      DC      0       SIN(    0       1      3   )
*TRAN      TSTEP  TSTOP  TSTART TMAX    ?UIC?
.tran      100m   1      0       1m
```

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

The points are spaced too close together to see any details.

```
=========================================================================
plot     v1
plot     v1    pointplot
```






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

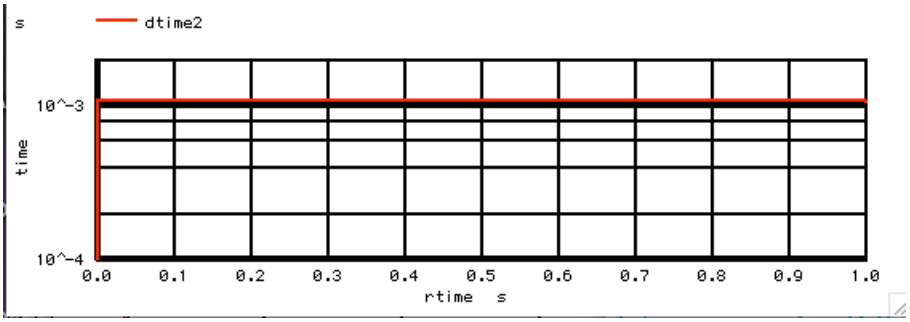
This is where being able to view timing in orders of magnitude comes in handy.

```
=========================================================================
let      num = length(time)-2
compose  dtime start = 0 stop = $&num step =1
compose  rtime start = 0 stop = $&num step =1
let      i = 0
repeat   $&num
let      i = i +1
let      dtime[i] = time[i +1] -time[i]
let      rtime[i] = time[i]
end
let      dtime2 = abs(dtime)+100u
plot     dtime2 vs rtime ylog
```

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

Except at the beginning, the sine wave appears to want to have a consistent timing. And the RMS value comes close.

```
=========================================================================
let      vrms1_cdhw = sqrt(mean(v1*v1))
echo     "INPUT SINE Square prelinear  =  $&vrms1_cdhw"
```



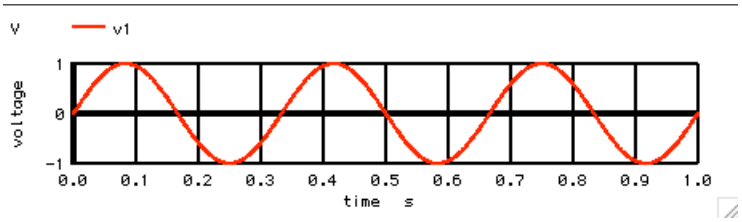

```
INPUT RMS SINE prelinear  =  0.70325
```

==========

The Linearize statement does not do much. But the RMS value comes closer still.

==========

```
linearize
plot     v1    pointplot
let      vrms1_cdhw = sqrt(mean(v1*v1))
echo    "INPUT SINE Square postlinear  =  $&vrms1_cdhw"
```




```
INPUT RMS SINE postlinear  =  0.706751
```

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

```
RMS_SINE_1msec
.Option srcsteps = 1 set Gmin = 1.0000E-02
*=========Circuit_Netlist=========================

V_SIN    V1    0       DC      0     SIN(    0    1 3   )
*TRAN    TSTEP TSTOP TSTART TMAX ?UIC?
.tran    1m     1      0       1m
.control
run
set      pensize = 2

plot     v1 pointplot
plot     v1
let      vrms1_cdhw = sqrt(mean(v1*v1))
echo     "INPUT RMS SINE prelinear  =  $&vrms1_cdhw"

let      num = length(time)-2
compose  dtime start = 0 stop = $&num step =1
compose  rtime start = 0 stop = $&num step =1
let      i = 0
repeat   $&num
let      i = i +1
let      dtime[i] = time[i +1] -time[i]
let      rtime[i] = time[i]
end
let      dtime2 = abs(dtime)+100u
plot     dtime vs rtime
plot     dtime2 vs rtime ylog


linearize
plot     v1 pointplot
plot     v1
let      vrms1_cdhw = sqrt(mean(v1*v1))
echo     "INPUT RMS SINE postlinear  =  $&vrms1_cdhw"

let      num = length(time)-2
```

```
compose  dtime start = 0 stop = $&num step =1
compose  rtime start = 0 stop = $&num step =1
let      i = 0
repeat   $&num
let      i = i +1
let      dtime[i] = time[i +1] -time[i]
let      rtime[i] = time[i]
end
plot     dtime vs rtime


.endc
.end
```

7.12.10_10.31AM
dsauersanjose@aol.com
Don Sauer
http://www.idea2ic.com/