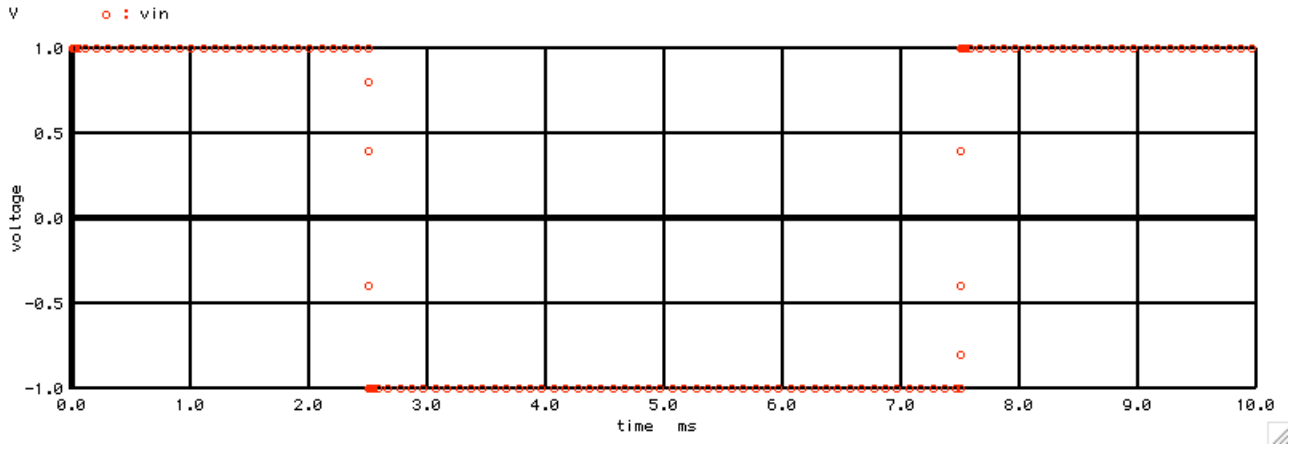
*=========TIMING_CHECK_OUT================

Spice tends to add more time points to a waveform when things are changing fast.

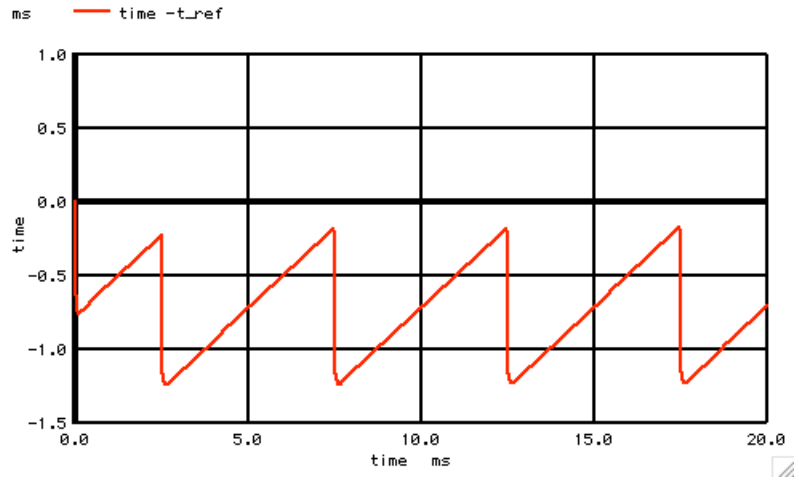
```
================================================================
*V_PULSE# NODE_P NODE_N DC     VALUE   PULSE( VINIT  VPULSE TDELAY TRISE  TFALL  PWIDTH PERIOD )
V_SQR     VIN    0      DC     0       PULSE( -1     1      -2.5m    1u     1u     5m     10m    )

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



Plotting a square wave as points can show the extra time points.

*=========TIMING_CHECK_OUT================

```
let       numb = length(vin)
print     numb
let       t_indx1 = vector($&numb)
let       t_ref   = vector($&numb)/$&numb
set       scale   t_indx1
plot      time -t_ref xlimit 0 20m
```

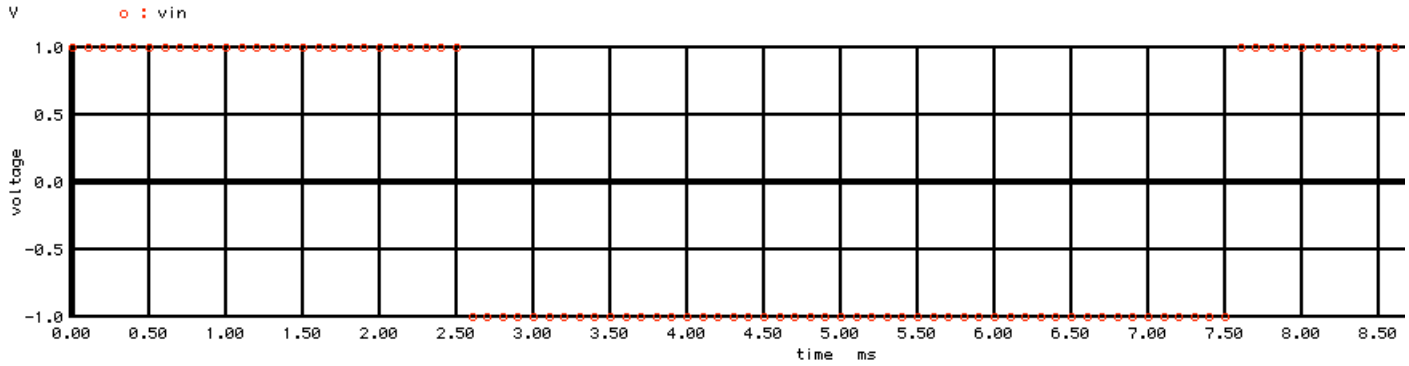



```
numb =  1.28110e+04
```

It just so happens that the transient is done for one sec with a maximum timing set to .1msec. One second at .1msec steps should give 10001 time points. Looks like another 2810 points got added.

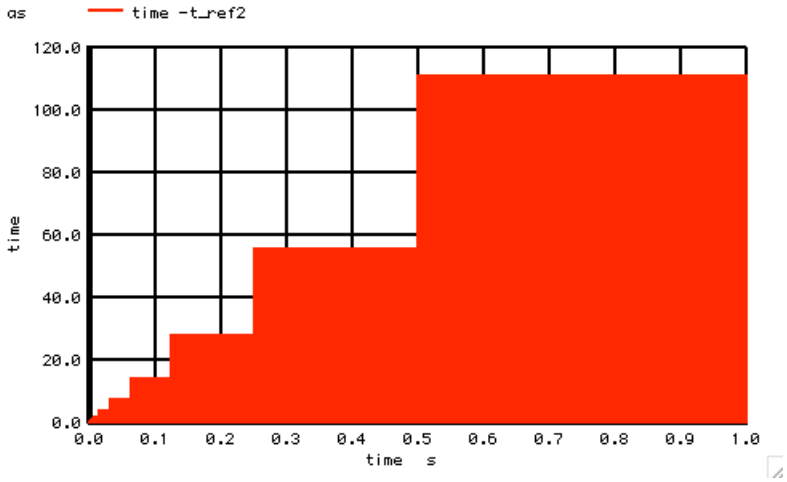
The vector function can be used to build a perfect timing ramp, which can be subtracted from the actual timing to see how the timing changes.

```
*=========LINEARIZE_TIMING================
linearize
plot          vin  xlimit 0 10m pointplot
```



Plotting a linearized square wave as points looks like consistent time points.

```
*=========LINEARIZE_TIMING_CHECK_OUT================
let       numb2 = length(vin)
print     numb2
let       numb3= numb2-1
let       t_indx2  = vector($&numb2)
let       t_ref2   = vector($&numb2)/$&numb3
set       scale t_indx2
plot      time -t_ref2
```




```
numb2 =   1.00010e+04
```

One second at .1msec steps should give 10001 time points. There should be one more time point and time periods. Two time periods at .1msec will need three time points to define the waveform from 0 to .2msec.

The actual difference between the perfect and actual timing appears to be very good.

```
===========Full_Netlist_For_Copy_Paste=======================

Check_Timing_Linearity
*V_PULSE# NODE_P NODE_N DC     VALUE  PULSE( VINIT  VPULSE TDELAY TRISE  TFALL  PWIDTH PERIOD )
V_SQR     VIN    0      DC     0      PULSE( -1     1      -2.5m    1u     1u     5m     10m    )

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

```
let       numb = length(vin)
print     numb
let       t_indx1 = vector($&numb)
let       t_ref   = vector($&numb)/$&numb
set       scale   t_indx1
plot      time -t_ref xlimit 0 20m


linearize
plot      vin  xlimit 0 10m pointplot
let       numb2 = length(vin)
print     numb2
let       numb3= numb2-1
let       t_indx2 = vector($&numb2)
let       t_ref2   = vector($&numb2)/$&numb3
set       scale t_indx2
plot      time -t_ref2


.endc
.end
```