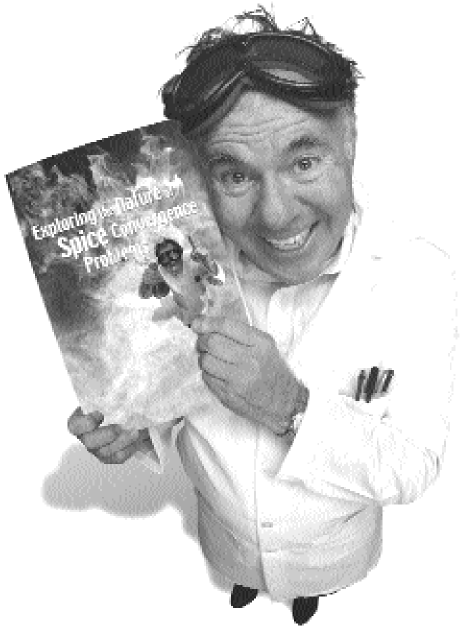

Exploring the Nature of

Spice Convergence Problems

Exploring the Nature of

Spice Convergence Problems

A discussion of Spice algorithms and guidlines for avoiding convergence problems.

table of contents

This appendix discusses common errors and convergence problems in PSpice.

Preface

Overcoming problems that impede convergence is key to successful Spice simulation. Some causes of convergence problems are common to all Spice simulators.

How to avoid them is explained here.

Note: PSpice® has several unique characteristics that make it better at achieving convergence than some other Spice simulators. Therefore, it is used in the discussion following.

Introduction

In order to calculate the bias point, DC sweep and transient analysis for analog devices PSpice must solve a set of nonlinear equations which describe the circuit's behavior. This is accomplished by using an iterative technique—the Newton-Raphson algorithm—which starts by having an initial approximation to the solution and iteratively improves it until successive voltages and currents converge to the same result.

In a few cases PSpice cannot find a solution to the nonlinear circuit equations. This is generally called a “convergence problem” because the symptom is that the Newton-Raphson repeating series cannot converge onto a consistent set of voltages and currents. The following discussion gives some background on the algorithms in PSpice and some guidelines for avoiding convergence problems.

The AC and noise analyses are linear and do not use an iterative algorithm, so the following discussion does not apply to them. Digital devices are evaluated using boolean algebra; this discussion does not apply to them either.

The transient analysis has the additional possibility of being unable to continue because the time step required becomes too small from something in the circuit moving too fast. This is also discussed herein.

Newton-Raphson requirements

The Newton-Raphson algorithm is **guaranteed to converge to a solution.** However, this guarantee has some conditions:

1 The nonlinear equations must have a solution.

2 The equations must be continuous.

3 The algorithm needs the equations' derivatives.

4 The initial approximation must be close enough to the solution.

Each of these can be taken in order. Remember that the PSpice algorithms are used in computer hardware that has finite precision and finite dynamic range that produce these limits:

- Voltages and currents in PSpice are limited to +/-1e10 volts and amps.

- Derivatives in PSpice are limited to 1e14.

- The arithmetic used in PSpice is double precision and has 15 digits of accuracy.

Is there a solution?

Yes, for any physically realistic circuit. However, it is not difficult to set up a circuit that has no solution within the limits of PSpice numerics. Consider, for example, a voltage source of one megavolt connected to a resistor of one micro-ohm. This circuit does not have a solution within the dynamic range of currents (+/- 1e10 amps). Here is another example:

```
V1      1,      0       5v
D1      1,      0       DMOD
.MODEL          DMOD(IS=1e-16)
```

The problem here is that the diode model has no series resistance.

To find out more about the diode equations, refer to the *Analog Devices* chapter in the online *PSpice A/D Reference Manual*.
It can be shown that the current through a diode is:

$$I = IS\ e^{V/(N\ k\ T)}$$

N defaults to one and k*T at room temperature is about .025 volts.
So, in this example the current through the diode would be:

$$I = 1e\text{-}16\ e^{200} = 7.22e70 \text{ amps}$$

This circuit also does not have a solution within the limits of the dynamic range of PSpice. In general, be careful of components without limits built into them. Extra care is needed when using the expressions for controlled sources (such as for behavioral modeling). It is easy to write expressions with very large values.

Are the Equations Continuous?

The device equations built into PSpice are continuous. The functions available for behavioral modeling are also continuous (there are several functions, such as int(x), which cannot be added because of this). So, for physically realistic circuits the equations can also be continuous. Exceptions that come are usually from exceeding the limits of the numerics in PSpice. This example tries to approximate an ideal switch using the diode model:

```
.MODEL DMOD(IS=1e-16 N=1e-6)
```

The current through this diode is:

$$I = 1e\text{-}16\ e^{V/(N\ .025)} = 1e\text{-}16\ e^{V/25e\text{-}9}$$

Avoid unrealistic model parameters. Behavioral modeling expressions need extra care.

Because the denominator in the exponential is so small, the current I is essentially zero for V < 0 and almost infinite for V > 0. Even if there are external components that limit the current, the "knee" of the diode's I-V curve is so sharp that it is almost a discontinuity.

Are the derivatives correct?

The device equations built into PSpice include the derivatives, and these are correct. Depending on the device, the physical meaning of the derivatives is small-signal conductance, transconductance or gain.

Unrealistic model parameters can exceed the limit of 1e14, but it requires some effort. The main thing to look at is the behavioral modeling expressions, especially those having denominators.

Is the initial approximation close enough?

Newton-Raphson is guaranteed to converge only if the analysis is started close to the answer. Also, there is no measurement that can tell how close is close enough.

PSpice gets around this by making heavy use of continuity. Each analysis starts from a known solution and uses a variable step size to find the next solution. If the next solution does not converge PSpice reduces the step size, falls back and tries again.

Bias point

The hardest part of the whole process is getting started, that is, finding the bias point. PSpice first tries with the power supplies set to 100%. A solution is not guaranteed, but most of the time the PSpice algorithm finds one. If not, then the power supplies are cut back to almost zero. They are cut to a level small enough that **all nonlinearities are turned off.** When the circuit is linear a solution can be found (very near zero, of course). Then, PSpice works its way back up to 100% power supplies using a variable step size.

Once a bias point is found the transient analysis can be run. It starts from a known solution (the bias point) and steps forward in time. The step size is variable and is reduced as needed to find further solutions.

DC sweep

The DC sweep uses a hybrid approach. It uses the bias point algorithm (varying the power supplies) to get started. For subsequent steps it uses the previous solution as the initial approximation. The sweep step is not variable, however. If a solution cannot be found at a step then the bias point algorithm is used for that step.

The whole process relies heavily on continuity. It also requires that the circuit be linear when the supplies are turned off.

STEPGMIN

An alterative algorithm is GMIN stepping. This is not obtained by default, and is enabled by specifying the circuit analysis option STEPGMIN (either using .OPTION STEPGMIN in the netlist, or by making the appropriate choice from the Analysis/Setup/Options menu). When enabled, the GMIN stepping algorithm is applied after the circuit fails to converge with the power supplies at 100 percent, and if GMIN stepping also fails, the supplies are then cut back to almost zero.

GMIN stepping attempts to find a solution by starting the repeating cycle with a large value of GMIN, initially 1.0e10 times the nominal value. If a solution is found at this setting it then reduces GMIN by a factor of 10, and tries again. This continues until either GMIN is back to the nominal value, or a repeating cycle fails to converge. In the latter case, GMIN is restored to the nominal value and the power supplies are stepped.

Bias point and DC sweep

Power supply stepping

As previously discussed, PSpice uses a proprietary algorithm which finds a continuous path from zero power supplies levels to 100%. It starts at almost zero (.001%) power supplies levels and works its way back up to the 100% levels. The minimum step size is 1e-6 (.0001%). The first repeating series of the first step **starts at zero for all voltages**.

Semiconductors

Model parameters

The first consideration for semiconductors is to avoid physically unrealistic model parameters. Remember that as PSpice steps the power supplies up it has to step carefully through the turn on transition for each device. In the diode example above, for the setting N=1e-6, the knee of the I-V curve would be too sharp for PSpice to maintain its continuity within the power supply step size limit of 1e-6.

Unguarded p-n junctions

A second consideration is to avoid “unguarded” p-n junctions (no series resistance). The above diode example also applies to the p-n junctions inside bipolar transistors, MOSFETs (drain-bulk and source-bulk), JFETs and GaAsFETs.

No leakage resistance

A third consideration is to avoid situations which could have an ideal current source pushing current into a reverse-biased p-n junction without a shunt resistance. Since p-n junctions in PSpice have (almost) no leakage resistance and would cause the junction's voltage to go beyond 1e10 volts.

The model libraries which are part of PSpice follow these guidelines.

Typos can cause unrealistic device parameters. The following MOSFET:

```
        M1 3, 2, 1, 0         MMOD L=5 W=3
```

has a length of five meters and a width of three meters instead of micrometers. It should have been:

```
        M1 3, 2, 1, 0         MMOD L=5u W=3u
```

PSpice flags an error for L too large, but cannot for W because power MOSFETs are so interdigitated (a zipper-like trace) that their effective W can be very high. The LIST option can show this kind of problem. When the devices are listed in the output file their values are shown in scientific notation making it easy to spot unusual values.

Switches

PSpice switches have gain in their transition region. If several are cascaded then the cumulative gain can easily exceed the derivative limit of 1e14. This can happen when modeling simple logic gates using totem-pole switches and there are several gates in cascaded in series. Usually a cascade of two switches works but three or more can cause trouble.

Behavioral modeling expressions

Range limits

Voltages and currents in PSpice are limited to the range +/- 1e10. Care must be taken that the output of expressions fall within this range. This is especially important when one is building an electrical analog of a mechanical, hydraulic or other type of system.

Source limits

Another consideration is that the controlled sources must turn off when the supplies are almost 0 (.001%). There is special code in PSpice which “squelches” the controlled sources in a continuous way near 0 supplies. However, care should still be taken using expressions that have denominators. Take, for example, a constant power load:

```
        GLOAD 3, 5     VALUE = {2Watts/V(3,5)}
```

The first repeating series starts with V(3,5) = 0 and the current through GLOAD would be infinite (actually, the code in PSpice which does the division clips the result to a finite value). The “squelching” code is required to be a smooth and well-behaved function.

Note: The “squelching” code cannot be “strong” enough to suppress dividing by 0.

The result is that GLOAD does not turn off near 0 power supplies. A better way is described in the application note Modeling Constant Power Loads. The “squelching” code is sufficient for turning off all expressions except those having denominators. In general, though, it is good practice to constrain expressions having the LIMIT function to keep results within physically realistic bounds.

Example: A first approximation to an opamp that has an open loop gain of 100,000 is:

```
        VOPAMP         3, 5   VALUE = {V(in+,in-)*1e5}
```

This has the undesirable property that there is no limit on the output. A better expression is:

```
        VOPAMP         3, 5   VALUE =
        + {LIMIT(V(in+,in-)*1e5,15v,-15v}
```

where the output is limited to +/- 15 volts.

Transient analysis

The transient analysis starts using a known solution - the bias point. It then uses the most recent solution as the first guess for each new time point. If necessary, the time step is cut back to keep the new time point close enough that the first guess allows the Newton-Raphson repeating series to converge. The time step is also adjusted to keep the integration of charges and fluxes accurate enough.

In theory the same considerations which were noted for the bias point calculation apply to the transient analysis. However, in practice they show up during the bias point calculation first and, hence, are corrected before a transient analysis is run.

The transient analysis can fail to complete if the time step gets too small. This can have two different effects:

1. The Newton-Raphson iterations would not converge even for the smallest time step size, or

2. Something in the circuit is moving faster than can be accommodated by the minimum step size.

The message PSpice puts into the output file specifies which condition occurred.

Skipping the bias point

The SKIPBP option for the transient analysis skips the bias point calculation. In this case the transient analysis has no known solution to start from and, therefore, is not assured of converging at the first time point. Because of this, its use is not recommended. Its inclusion in PSpice is to maintain compatibility with UC Berkeley SPICE. SKIPBP has the same meaning as UIC in Berkeley SPICE. UIC is not needed in order to specify initial conditions.

The dynamic range of TIME

TIME, the simulation time during transient analysis, is a double precision variable which gives it about 15 digits of accuracy. The dynamic range is set to be 15 digits minus the number of digits of accuracy required by RELTOL. For a default value of RELTOL = .001 (.1% or 3 digits) this gives 15-3 = 12 digits. This means that the minimum time step is the overall run time (TSTOP) divided by 1e12. The dynamic range is large but finite.

It is possible to exceed this dynamic range in some circuits. Consider, for example, a timer circuit which charges up a 100uF capacitor to provide a delay of 100 seconds. At a certain threshold a comparator turns on a power MOSFET. The overall simulation time is 100 seconds. For default RELTOL this gives us a minimum time step of 100 picoseconds. If the comparator and other circuitry has portions that switch in a nanosecond then PSpice needs steps of less than 100 picoseconds to calculate the transition accurately.

Failure at the first time step

If the transient analysis fails at the first time point then usually there is an unreasonably large capacitor or inductor. Usually this is due to a typographical error. Consider the following capacitor:

```
        C 1 3, 0 1Ouf
```

“1O” (has the letter O) should have been “10.” This capacitor has a value of one farad, not 10 microfarads. An easy way to catch these is to use the LIST option (on the .OPTIONS command).

LIST

The LIST option can echo back all the devices into the output file
that have their values in scientific notation.

That makes it easy to spot any unusual values. This kind of problem does not show up during the bias point calculation because capacitors and inductors do not participate in the bias point.

Similar comments apply to the parasitic capacitance parameters in transistor (and diode) models. These are normally echoed to the output file (the NOMOD option suppresses the echo but the default is to echo). As in the LIST output, the model parameters are echoed in scientific notation making it easy to spot unusual values. A further diagnostic is to ask for the detailed operating bias point (.TRAN/OP) information.

.TRAN/OP

This lists the small-signal parameters for each semiconductor device including the calculated parasitic capacitances.

Parasitic capacitances

It is important that switching times be nonzero. This is assured if devices have parasitic capacitances. The semiconductor model libraries in PSpice have such capacitances. If switches and/or controlled sources are used, then care should be taken to assure that no sections of circuitry can try to switch in zero time. In practice this means that if any positive feedback loops exist (such as a Schmidt trigger built out of switches) then such loops should include capacitances.

Another way of saying all this is that during transient analysis the circuit equations must be continuous over time (just as during the bias point calculation the equations must be continuous with the power supply level).

Inductors and transformers

While the impedance of capacitors gets lower at high frequencies (and small time steps) the impedance of inductors gets higher.

Note: The inductors in PSpice have an infinite bandwidth.

Real inductors have a finite bandwidth due to eddy current losses and/or skin effect. At high frequencies the effective inductance drops. Another way to say this is that physical inductors have a frequency at which their Q begins to roll off. The inductors in PSpice have no such limit. This can lead to very fast spikes as transistors (and diodes) connected to inductors turn on and off. The fast spikes, in turn, can force PSpice to take unrealistically small time steps.

Note: Orcad recommends that all inductors have a parallel resistor (series resistance is good for modeling DC effects but does not limit the inductor’s bandwidth).

The parallel resistor gives a good model for eddy current loss and limits the bandwidth of the inductor. The size of resistor should be set to be equal to the inductor’s impedance at the frequency at which its Q begins to roll off.

Example: A common one millihenry iron core inductor begins to roll off at no less than 100KHz. A good resistor value to use in parallel is then R = 2*π*100e3*.001 = 628 ohms. Below the roll-off frequency the inductor dominates; above it the resistor does. This keeps the width of spikes from becoming unreasonably narrow.

Bipolar transistors substrate junction

The UC Berkeley SPICE contains an unfortunate convention for the substrate node of bipolar transistors. The collector-substrate p-n junction has no DC component. If the capacitance model parameters are specified (e.g., CJS) then the junction has (voltage-dependent) capacitance but no DC current. This can lead to a sneaky problem: if the junction is inadvertently forward-biased it can create a very large capacitance. The capacitance goes as a power of the junction voltage. Normal junctions cannot sustain much forward voltage because a large current flows. The collector-substrate junction is an exception because it has no DC current.

If this happens it usually shows up at the first time step. It can be spotted turning on the detailed operating point information (.TRAN/OP) and looking at the calculated value of CJS for bipolar transistors. The whole problem can be prevented by using the PSpice model parameter ISS. This parameter “turns on” DC current for the substrate junction.

Diagnostics

If PSpice encounters a convergence problem it inserts into the output file a message that looks like the following.

```
ERROR -- Convergence problem in transient analysis at Time = 7.920E-03

Time step =  47.69E-15, minimum allowable step size = 300.0E-15

These voltages failed to converge:
         V(x2.23)  =          1230.23 / -68.4137
                  V(x2.25)  =        -1211.94 / 86.6888

These supply currents failed to converge:
         I(X2.L1)   =         -36.6259 / 2.25682
                  I(X2.L2)   =         -36.5838 / 2.29898

These devices failed to converge:
         X2.DCR3   X2.DCR4   x2.ktr    X2.Q1      X2.Q2

Last node voltages tried were:
NODE      VOLTAGE     NODE     VOLTA GE  NODE     VOLTAGE    NODE       VOLTAGE
(  1)     25.2000     (  3)    4.0000    (  4)    0.0000     (  6)      25.2030
(x2.23)   1230.2000   (X2.24)  9.1441    (x2.25)  -1211.9000 (X2.26)    256.9700
(X2.28)   -206.6100   (X2.29)  75.4870   (X2.30)  -25.0780   (X2.31)    26.2810
(X3.34)   1.771E-06   (X3.35)  1.0881    (X3.36)  .4279      (X2.XU1.6) 1.2636
```

The message always includes the banner (`ERROR -- convergence problem ...`) and the trailer (`Last node voltages tried were ...`). It cannot include all three of the middle blocks.

The `Last node voltages tried...` trailer shows the voltages tried at the last Newton-Raphson iteration. If any of the nodes have unreasonable large values this is a clue that these nodes are related to the problem. `These voltages failed to converge` lists the specific nodes which did not settle onto consistent values. It also shows their values for the last two iterations. These `supply currents failed converge` does the same for currents through voltage sources and inductors. If any of the listed numbers are +/- 1e10 then that is an indication that the value is being clipped from an unreasonable value. Finally, `These devices failed to converge` shows devices whose terminal currents or core fluxes did not settle onto consistent values.

The message gives a clue as to the part of the circuit which is causing the problem. Looking at those devices and/or nodes for the problems discussed above is recommended.