Solving SPICE Convergence Problems

Key Sources

The following techniques on solving convergence problems are taken from various sources including:

[1] Meares, L.G., Hymowitz C.E. “SIMULATING WITH SPICE”, Intusoft, 1988
[2] Muller, K.H. ”A SPICE COOKBOOK”, Intusoft, 1990
[3] Meares, L.G., Hymowitz C.E. “SPICE APPLICATIONS HANDBOOK”, Intusoft, 1990
[4] *Intusoft Newsletters*, various dates from 1986 to present
[5] The Designer's Guide to SPIC and Spectre, Kenneth S. Kundert, Kluwer Academic Publishers, 1995
[6] The SPICE Book, Andrei Vladimirescu, John Wiley & Sons Inc,, 1994
[7] Inside SPICE, Ron Kielkowski, McGraw-Hill, Inc. 1994

What is Convergence? **(or in my case, Non-Convergence)**

The answer to a nonlinear problem, such as those in the SPICE DC and Transient analyses, is found via an iterative solution. For example, ISSPICE makes an initial guess at the circuit’s node voltages and then, using the circuit conductances, calculates the mesh currents. The currents are then used to recalculate the node voltages, and the cycle begins again. This continues until all of the node voltages settle to values which are within specific tolerance limits. These limits can be altered using various .Options parameters such as Reltol, Vntol, and Abstol.

If the node voltages do not settle down within a certain number of iterations, the DC analysis will issue an error message such as *"No convergence in DC analysis"*, *"Singular Matrix"*, or *"Gmin/Source Stepping Failed"*. SPICE will then terminate the run because both the AC and transient analyses require an initial stable operating point in order to proceed. During the transient analysis, this iterative process is repeated for each individual time step. If the node voltages do not settle down, the time step is reduced and SPICE tries again to determine the node voltages. If the time step is reduced beyond a specific fraction of the total analysis time, the transient analysis will issue the error message, *"Time step too small,"* and the analysis will be halted.

Problems come in all shapes, sizes, and disguises, but convergence problems are usually related to one of the following:

- **Circuit Topology**
- **Device Modeling**
- **Simulator Setup**

The DC analysis may fail to converge because of incorrect initial voltage estimates, model discontinuities, unstable/bistable operation, or unrealistic circuit impedances. Transient analysis failures are usually due to model discontinuities or unrealistic circuit, source, or parasitic modeling. In general, you will have problems if the impedances, or impedance changes, do not remain reasonable. Convergence problems will result if the impedances in your circuit are too high or too low.

The various solutions to convergence problems fall under one of two types. Some are simply band-aids which merely attempt to fix the symptom by adjusting the simulator options. Other solutions actually affect the true cause of the convergence problems.

The following techniques can be used to solve 90 to 95% of all convergence problems. When a convergence problem is encountered, you should start at solution 0 and proceed with the subsequent suggestions until convergence is achieved. The sequence of the suggestions is structured so that they can be incrementally added to the simulation. The sequence is also defined so that the initial suggestions will be of the most benefit. Note that suggestions which involve simulation options may simply mask the underlying circuit instabilities. Invariably, you will find that once the circuit is properly modeled, many of the "options" fixes will no longer be required!

General Discussion

Many power electronics convergence problems can be solved with two option parameters, Gmin and Rshunt. The Gmin option is available in all SPICE 2 and 3 programs. Gmin is the minimum conductance across all semiconductor junctions. The conductance is used to keep the matrix well conditioned. Its default value is 1E-12mhos. Setting Gmin to a value between 1n and 10n will often solve convergence problems. Setting Gmin to a value which is greater than 10n may cause convergence problems.

The Rshunt option causes IsSpice to insert a resistor from every node in the circuit to ground. Rshunt is available only in programs such as IsSpice that have incorporated the XSPICE enhancements [36]. Setting Rshunt to a value between 100MEG and 1G will typically help. Setting Rshunt to a value of 100K may cause convergence problems.

SPICE does not always converge when relaxed tolerances are used. One of the most common problems is the incorrect use of the .Options parameters. For example, setting the tolerance option, Reltol, to a value which is greater than .01 will often cause convergence problems.

The default numerical integration method is the Trapezoidal method. Some circuits will converge better during the transient analysis when the Gear integration method is used. You can invoke Gear integration by adding the statement, .OPTIONS METHOD=GEAR. The Gear method works well for most power electronics simulations.

Setting the value of Abstol to 1u will help in the case of circuits that have currents which are larger than several amps. Again, do not overdo this option. Setting Abstol to a value which is greater than 1u will cause more convergence problems than it will solve.

After you've performed a number of simulations, you will discover the options which work best for your circuits. You can save the .Options line in a text file and use the ".INC filename" command to read the text file. This will allow you to save several .Options lines in text files and explore the use of different sets of options.

If all else fails, you can almost always get a circuit to simulate in a transient simulation if you begin with a zero voltage/zero current state. This makes sense if you consider the fact that the simulation always starts with the assumption that all voltages and currents are zero. The simulator can almost always track the nodes from a zero condition. Running the simulation will often help uncover the cause of the convergence failure.

The above recommendation is only true if your circuit is constructed properly and the netlist is syntactically correct. Most of the time, minor mistakes are the cause of convergence problems. Error messages will help you track down the problems, however, a good technique is to scan each line of the netlist and look for anomalies. It may be tedious, but it's a proven way to weed out mistakes.

Not all convergence failures are a result of the SPICE software! Convergence failures may identify many circuit problems. Check your circuits carefully, and don't be too quick to blame the software.

If you've tried everything you can think of, and you still can't get your circuit to converge, you may contact Intusoft's Technical Support staff at info@intusoft.com We can also be reached on the Internet at http://www.intusoft.com.

ISSPICE - New Convergence Algorithms

In addition to automatically invoking the traditional source stepping algorithm, ISSPICE contains a superior algorithm called "Gmin Stepping". This algorithm uses a constant minimal junction conductance which keeps the sparse matrix well conditioned, and a separate variable conductance to ground at each node, which serves as a DC convergence aid. The variable conductances cause the solution to converge more quickly. They are then reduced, and the solution is re-computed. The solution is eventually found with a sufficiently small conductance. Then the conductance is removed entirely in order to obtain a final solution. This technique has proven to work very well, and ISSPICE selects it automatically when convergence problems occur. The suggestion (made in a number of textbooks) of increasing the .Options Gmin value in order to solve DC and operating point convergence problems is performed automatically by this new algorithm. Gmin may still be increased (relaxed) for the entire simulation by setting the .Options Gmin value, but this should only be done as a last resort.

Non-Convergence Error Messages/Indications

The following is a list of the key error messages which indicate that convergence has not occurred. In most cases, SPICE 3 will also indicate the element or node that is the source of the failure. This is a feature which is not found in most other SPICE 2 simulators.

- DC Analysis (which includes the .OP analysis and the small signal bias solution which is performed prior to the AC analysis or Initial transient solution which is calculated prior to the Transient analysis) - *“No Convergence in DC analysis”*, or *“PIVTOL Error”*. SPICE 3 programs such as ISSPICE issue a *“Gmin/Source Stepping Failed”* or *“Singular Matrix”* message.

- DC Sweep Analysis (.DC) - *“No Convergence in DC analysis at Step = xxx.”*
- Transient Analysis (.TRAN) - *“Internal timestep too small.”*

Convergence Solutions

Important Note: The suggestions below are applicable to most SPICE programs, especially if they are Berkeley SPICE 3 compatible. If several .Options parameters are used, you can put them on the same line and separate them with spaces.

DC Convergence Solutions

0. Check the circuit topology and connectivity.
“.Options LIST” will provide a nice summary printout of the nodal connections. It produces a flattened netlist of the entire circuit in the output file. If you are using the SPICENET schematic entry program, you should perform a “ReNet” to insure that unique node numbers and reference designations are being used and that all circuit elements are properly connected.

Common mistakes and problems:

- Make sure that all of the circuit connections are valid. Check for incorrect node numbering or dangling nodes. Also, verify component polarity.
- Make sure you didn’t use the letter O instead of a zero (0).
- Check for syntax mistakes. Make sure that you used the correct SPICE units (i.e. MEG instead of M(milli) for 1E6).
- Make sure that there’s a DC path from every node to ground.
- Make sure that there are at least two connections at every node.

- Make sure that there are no loops of inductors or voltage sources.
- Make sure that there are no series capacitors or current sources.
- Place the ground (node 0) somewhere in the circuit. Be careful when you use floating grounds; you may need to connect a large resistor from the floating node to ground.
- Make sure that voltage/current generators use realistic values, and verify that the syntax is correct.
- Make sure that dependent source gains are correct, and that B element expressions are reasonable. If you are using division in an expression, verify that division by zero cannot occur.
- Make sure that there are no unrealistic model parameters; especially if you have manually entered the model into the netlist.
- Make sure that all resistors have a value. In SPICE 3, resistors without values are given a default value of 1 kOhm.
- Negative capacitor and inductor values are allowed in SPICE 3. They will not be flagged as an error, but can cause timestep problems, depending on the topology of the circuit.

1. Increase ITL1 to 400 in the .OPTIONS statement.

Example: .OPTIONS ITL1=400

This increases the number of DC iterations that IsSpice will perform before it gives up. In all but the most complex circuits, further increases in ITL1 won't typically aid convergence.

2. Set ITL6 =100 in the .OPTIONS statement. (ITL6 is only used for SPICE 2 based simulators). Srcsteps is used for SPICE 3 simulators.

Example: .OPTIONS SRCSTEPS=100

This invokes the source stepping algorithm. The value is the number of steps. This solution is unnecessary for IsSpice users, since source stepping is automatically invoked after both the default method and the Gmin stepping algorithms have been attempted. Note for SPICE 2 users: this is an undocumented Berkeley SPICE 2G option.

Source stepping sets all of the stimulus functions (voltage sources, etc.) to a near zero value in the hopes of easing the calculation of the operating point solution. When a solution is found, the stimulus sources are increased toward their final DC values, and another operating point is calculated using the previous solution as a “seed”. This process continues until the sources are at the full DC values and an operating point in produced.

3. Add .NODESETs

Example: .NODESET V(6)=0

View the node voltage/branch current table in the output file. SPICE 3 produces one even if the circuit does not converge. Add .NODESET values for the top level circuit nodes (not the subcircuit nodes) that have unrealistic values. You do not need to nodeset every node. Use a .NODESET value of 0V if you do not have a better estimation of the proper DC voltage. Caution is warranted, however, for an inaccurate Nodeset value may cause undesirable results.

4. Add resistors and use the OFF keyword.

```
Example:     D1   1  2  DMOD OFF
             RD1  1  2  100MEG
```

Add resistors across diodes in order to simulate leakage. Add resistors across MOSFET drain-to-source connections to simulate realistic channel impedances. This will make the impedances reasonable so that they will be neither too high nor too low. Add ohmic resistances (RC, RB, RE) to transistors. Use the .Options statement to Reduce Gmin by an order of magnitude.

Next, you can also add the OFF keyword to semiconductors (especially diodes) that may be causing convergence problems. The OFF keyword tells ISSPICE to first solve the operating point with the device turned off. Then the device is turned on, and the previous operating point is used as a starting condition for the final operating point calculation.

5. Use PULSE statements to turn on DC power supplies.

Example: VCC 1 0 15 DC

becomes VCC 1 0 PULSE 0 15

This allows the user to selectively turn on specific power supplies. This is sometimes known as the “Pseudo-Transient” start-up method. Use a reasonable rise time in the PULSE statement to simulate realistic turn on. For example,

V1 1 0 PULSE 0 5 0 1U

will provide a 5 volt supply with a turn on time of 1 μs. The first value after the 5 (in this case, 0) is the turn-on delay, which can be used to allow the circuit to stabilize before the power supply is applied.

6. Set RSHUNT=xxx in the .OPTIONS statement.

Example: .OPTIONS RSHUNT=100MEG

The Rshunt option places a resistor, of the specified value, from every node in the circuit to ground. Note: if this works, you have indeed changed the operation of the circuit, so make sure that you verify the results carefully.

7. Add UIC (Use Initial Conditions) to the .TRAN statement.

Example: .TRAN .1N 100N UIC

Insert the UIC keyword in the .TRAN statement. Use Initial Conditions (UIC) will cause SPICE to completely bypass the DC analysis. You should add any applicable .IC and IC= initial conditions statements to assist in the initial stages of the transient analysis. Be careful when you set initial conditions, for a poor setting may cause convergence difficulties.

AC Analysis Note: Solutions 5 through 7 should be used only as a last resort, because they will not produce a valid DC operating point for the circuit (all supplies may not be turned On and circuit may not be properly biased). Therefore, you cannot use solutions 5-7 if you want to perform an AC analysis, because the AC analysis must be proceeded by a valid operating point solution. However, if your goal is to proceed to the transient analysis, then solutions 5-7 may help you and may possibly uncover the hidden problems which plague the DC analysis.

DC Sweep Convergence Solutions

0. Check circuit topology and connectivity.
This item is the same as item 0 in the DC analysis.

1. Set ITL2=100 in the .OPTIONS statement.

```
Example: .OPTIONS ITL2=100
```

This increases the number of DC iterations that SPICE will attempt before it gives up.

2. Increase or decrease the step values which are used in the .DC sweep.

```
Example: .DC VCC  0  1  .1
becomes  .DC VCC  0  1  .01
```

Discontinuities in the SPICE models can cause convergence problems. The use of larger steps may help to bypass the discontinuities, while the use of smaller steps may help IsSpice find the intermediate answers which will be used to find the point which doesn't converge.

3. Do not use the DC sweep analysis.

```
Example:     .DC VCC  0  5  .1
             VCC  1  0
becomes      .TRAN .01  1
             VCC  1  0  PULSE  0  5  0  1
```

In many cases, it is preferable to use the transient analysis to ramp the appropriate voltage and/or current sources. The transient analysis tends to be more robust, and is sometimes faster.

Transient Convergence Solutions

0. Check circuit topology and connectivity.
This item is the same as item 0 in the DC analysis.

1. Set RELTOL=.01 in the .OPTIONS statement.

```
Example: .OPTIONS RELTOL=.01
```

This option is encouraged for most simulations, since the reduction of Reltol can increase the simulation speed by 10 to 50%. Only a minor loss in accuracy usually results. A useful recommendation is to set Reltol to .01 for initial simulations, and then reset it to its default value of .001 when you have the simulation running the way you like it and a more accurate answer is required. Setting Reltol to a value less than .001 is generally not required.

2. Reduce the accuracy of ABSTOL/VNTOL if current/ voltage levels allow it.

Example: . OPTION ABSTOL=1N VNTOL=1M

Abstol and Vntol should be set to about 8 orders of magnitude below the level of the maximum voltage and current. The default values are Abstol=1pA and Vntol=1μV. These values are generally associated with IC designs.

3. Set ITL4=500 in the .OPTIONS statement.

Example: .OPTIONS ITL4=500

This increases the number of transient iterations that SPICE will attempt at each time point before it gives up. Values which are greater than 500 won’t usually bring convergence.

4. Realistically Model Your Circuit; add parasitics, especially stray/junction capacitance.

The idea here is to smooth any strong nonlinearities or discontinuities. This may be accomplished via the addition of capacitance to various nodes and verifying that all semiconductor junctions have capacitance. Other tips include:

- Use RC snubbers around diodes.
- Add Capacitance for all semiconductor junctions (3pF for diodes, 5pF for BJTs if no specific value is known).
- Add realistic circuit and element parasitics.
- Watch the Real-time display (If you have IsSpice) and look for waveforms that transition vertically (up or down) at the point during which the analysis halts. These are the key nodes which you should examine for problems.
- If the .Model definition for the part doesn’t reflect the behavior of the device, use a subcircuit representation.

This is especially important for RF and power devices such as RF BJTs and power MOSFETs. Many vendors cheat and try to “force fit” the SPICE .MODEL statement in order to represent a device’s behavior. This is a sure sign that the vendor has skimped on quality in favor of quantity. Primitive .MODEL statements CAN NOT be used to model most devices above 200MEGHz because of the effect of package parasitics. And .MODEL statements CAN NOT be used to model most power devices because of eir extreme nonlinear behavior. In particular, if your vendor uses a .MODEL statement to model a power MOSFET, throw away the model. It’s almost certainly useless for transient analysis.

5. Reduce the rise/fall times of the PULSE sources.

```
Example:     VCC  1  0  PULSE  0  1  0  0   0
becomes      VCC  1  0  PULSE  0  1  0  1U  1U
```

Again, we are trying to smooth strong nonlinearities. The pulse times should be realistic, not ideal. If no rise or fall time values are given, or if 0 is specified, the rise and fall times will be set to the TSTEP value in the .TRAN statement.

6. Use the .OPTIONS RAMPTIME=xxx statement to ramp up all of the sources.

```
Example: .OPTIONS RAMPTIME=10NS
```

Ramptime causes all the independent sources to be ramped up from zero to their initial values at the beginning of the transient analysis. The time is specified by the user. This may be quite helpful if you’re having trouble getting the transient analysis to start. Remember to give enough time for the sources to ramp up. If a ramp time is too short, it may cause disturbances that require a long time to settle, or may even cause further convergence problems.

7. Add UIC (Use Initial Conditions) to the .TRAN line.

```
Example: .TRAN .1N 100N UIC
```

If you are having trouble getting the transient analysis to start because the DC operating point can’t be calculated, insert the UIC keyword in the .TRAN statement. UIC will cause SPICE to completely bypass the DC analysis. You should add any applicable .IC and IC= initial conditions statements to assist in the initial stages of the transient analysis. Be careful when you set initial conditions, for a poor setting may cause convergence difficulties. (See the Altinit and Ramptime options for more help with UIC cases).

8. Change the integration method to Gear (See also Special Cases below).

Example: .OPTIONS METHOD=GEAR

This option causes SPICE 3 to use Gear integration to solve the transient equations, as opposed to the default method of trapezoidal integration. The use of the Gear integration method should be coupled with a reduction in the Reltol value. This will produce answers which approach a more stable numerical solution. Trapezoidal integration tends to produce a less stable solution which can produce spurious oscillations. Gear integration often produces superior results for power circuitry simulations, due to the fact that high frequency ringing and long simulation periods are often encountered.

Gear integration is very valuable, especially for Power Supply designers. It is included in all IsSpice versions. Many popular versions of SPICE, including Pspice™, Hspice™ and Electronics Workbench™ do NOT let you set this valuable and important option.

Modeling Tips

Device modeling is one of the hardest steps encountered in the circuit simulation process. It requires not only an understanding of the device's physical and electrical properties, but also a detailed knowledge of the particular circuit application. Nevertheless, the problems of device modeling are not insurmountable. A good first-cut model can be obtained from data sheet information and quick calculations, so the designer can have an accurate device model for a wide range of applications.

Data sheet information is generally very conservative, yet it provides a good first-cut of a device model. In order to obtain the best results for circuit modeling, follow the rule: “Use the simplest model possible”. In general, the SPICE component models have default values that produce reasonable first order results. Here are some helpful tips:

- Don’t make your models any more complicated than they need to be. Overcomplicating a model will only cause it to run more slowly, and will increase the likelihood of an error.
- Remember: modeling is a compromise.
- Don’t be afraid to test your models, especially the ones you did not create.
- Create subcircuits which can be run and debugged independently. Simulation is just like being at the bench. If the simulation of the entire circuit fails, you should break it apart and use simple test circuits to verify the operation of each component or section.
- Document the models as you create them. If you don’t use a model often, you might forget how to use it.
- Be careful when you models which have been produced by hardware vendors. Many have syntactical errors, and certainly DO NOT fully reflect the characteristics of the real part. Check the documentation for a list of characteristics which are supported by the model.
- Semiconductor models should always include junction capacitance and the transit time (AC charge storage) parameters.
- If the .Model definition for a large geometry device doesn’t reflect the behavior of the device, use a subcircuit representation.
- Be careful when using behavioral models for power devices. Many SPICE vendors try to pass off power semiconductor models using behavioral modeling techniques. Most SPICE vendors do not have the expertise to create sophisticated subcircuit representations. Behavioral models have their place, but in the case of power devices, they will usually NOT exhibit many important second order effects.

- **And lastly, there is no substitute for knowing what you're doing!!**

Intusoft makes available an inexpensive modeling program. The program, called SPICEMOD, is an easy-to-use utility that makes semiconductor models (Diode, Zener Diode, BJT, Power BJT, Darlington BJT, MOSFET, Power MOSFET, JFET, Triac, IGBT, SCR) from data sheet parameters. The models work with ANY SPICE simulator. It has two distinct advantages:

1) It allows you to make a SPICE model based on your design specifications. For example, you can make a model for 1A 100V diode. You can then simulate your circuit and refine the boundaries for the type of part required. You can assign the actual part number at a later time. This eliminates the need for your SPICE vendor to supply models for every possible part number.

2) Models are created from data sheet values. If you do not have all of the parameters, SpiceMod will estimate the data you do not have, based on the data you do have. Therefore, it never leaves key SPICE parameters (capacitance, transit time, etc.) at their default values. The use of these default values is the simplest way to make a good model useless.

SPICEMOD is highly recommended.

Repetitive And Switching Simulations

Switching simulations refer to simulations which have a significant number of repetitive cycles, such as those found in SMPS simulations. Simulations such as these can experience a large number of rejected timepoints. Rejected timepoints are due to the fact that SPICE has a dynamically varying timestep which is controlled by constant tolerance values (Reltol, Abstol, Vntol). An event that occurs during each cycle, such as the switching of a power semiconductor, can trigger a reduction in the timestep value. This is caused by the fact that SPICE attempts to maintain a specific accuracy, and adjusts the timestep in order to accomplish this task. The timestep is increased after the event, until the next cycle, when it is again reduced. This timestep hysteresis can cause an excessive number of unnecessary calculations. To correct this problem, we can regress to a SPICE version 1 methodology and force the simulator to have a fixed timestep value.

To force the timestep to be a fixed value, set the Trtol value to 100, i.e. .OPTIONS TRTOL=100. The default value is 7. The Trtol parameter controls how far ahead in time SPICE tries to jump. The value of 100 causes SPICE to try to jump far ahead. Then set the Tmax value in the .TRAN statement to a value which is between 1/10 and 1/100 of the switching cycle period (.TRAN tStep tStop tStart TMAX). This has the opposite effect; it forces the timestep to be limited. Together, they effectively lock the simulator timestep to a value which is between 1/10 and 1/100 of the switching cycle period, and eliminate virtually all of the rejected timepoints. These settings can result in over a 100% increase in speed!

Note: In order to verify the number of accepted and rejected timepoints, you may issue the .OPTIONS ACCT parameter and view the data at the end of the output file.

Other Convergence Helpers

For those users who are using a version of SPICE which is based on Berkeley SPICE 3, such as ISSPICE, several other options are also available:

1. Gminsteps (DC Convergence)

Example: .OPTIONS GMINSTEPS=200

The Gminsteps option adjusts the number of Gmin increments that will be used during the DC analysis. Gmin stepping is invoked automatically when there is a convergence problem. Gmin stepping is a new algorithm in SPICE 3 that greatly improves DC convergence.

2. ALTINIT function (Transient Convergence with UIC)

Example: . OPTION ALTINIT=10

Setting Altinit to 1 causes an alternate (more lenient) algorithm to be used when the UIC keyword is issued in the .TRAN statement. Normally, this alternate algorithm is automatically invoked when the default method fails. A number other than 1 refers to the initial timestep jump which will be used to determine the first timepoint. The default value is 1E-20 seconds. It can be varied from 1E-10 to 1E-30 seconds. The value of 1E-10 (i.e. Altinit=10) will reduce the accuracy of the first timepoint, but will make it easier for IsSpice to start the transient simulation. The Altinit option is unique to IsSpice.

Special Cases

Mosfets - Check the connectivity. Connecting two gates together, but to nothing else, will give a PIVTOL/Singular matrix error. Check the model Level parameter. SPICE 2 programs do not behave properly when Mosfets of different levels are used in the same simulation.

SPICE 3 Convergence Helpers

For those users who are running a version of SPICE based on Berkeley SPICE 3, several other options are also available.

1. **Gminsteps (DC Convergence) - Same as ITL6**
   Example: .OPTIONS GMINSTEPS=200

The Gminsteps option adjusts the number of increments that Gmin will be stepped during the DC analysis. Gmin stepping is invoked automatically when there is a convergence problem. Gmin stepping is a new algorithm in IsSpice4 that greatly improves DC convergence.

2. **ALTINIT function (Transient Convergence)**
   Example:.OPTIONS ALTINIT=1

Setting ALTINIT to one causes the default algorithm used when the UIC (use initial condition) keyword is issued in the .TRAN to be bypassed in favor of a second more lenient algorithm. Normally, the second algorithm is automatically invoked when the default method fails.