Is My Solution Converged Well Enough?

From OpenDSSWiki

Question

How can I tell if my solution is converged well enough to trust the currents computed in some of the smaller branches?

Answer

Since the OpenDSS solves for the node voltages first and then solves for the currents by feeding the voltage solution back to each individual circuit element, there is a possibility the reported current could contain some error. If circuit element impedance is too small (admittance very large), you can have a situation where the difference in the voltages is very tiny, which loses some accuracy, and then gets multiplied by a large number. The voltages are frequently accurate enough for distribution analysis, but the reported currents could be off. It is frequently obvious due to a very large reported current or power.

There are two "Show" commands that can help diagnose this problem:

```
Show Convergence
Show Mismatch
```

Show Convergence shows the convergence achieved in the node voltages for the most recent Solve command. If the voltage base for the node is provided, the convergence is stated in per unit. The default voltage tolerance is 0.0001.

Show Mismatch shows the sum of the currents at each node in both actual amperes and in percent of the current in circuit elements connected to the bus. The currents should be very small in both amperes and percent.

See the Word Doc file "Computing Element Currents in OpenDSS.Doc" in the Doc folder:

OpenDSS Doc Folder (http://electricdss.svn.sourceforge.net/viewvc/electricdss/Doc/)

Correcting for Excessive Mismatch

Sometimes, when operating the OpenDSS interactively, you can simply execute another Solve command. The OpenDSS with do at least two more iterations, which will generally improve the accuracy of the voltage solution.

Of course, this is not practical for Daily or Yearly solutions. To force more iterations, specify a tighter convergence tolerance.

If the problem persists, the remaining solution is to increase the impedance of the tiny line sections or simply remove them and alter bus connection accordingly. The problem usually arises when a data set from another computer program contains zero impedance line sections to represent switches or busbar sections. Since the OpenDSS is an admittance formulation, it is not possible to represent zero-impedance branches. The program does use double-precision floating-point math throughout, with only a few exceptions. Therefore, it is possible to have quite a large range of impedance and voltage values without precision problems.

There is some error-trapping for zero-impedance branches, and we continue to add more as we discover problems, but it is still possible to sneak one in. It will generally manifest itself as a floating-point overflow or divide-by-zero. The OpenDSS will generally not shut down for such errors since they are trapped. However, you may need to clear and re-compile the circuit script.

The best solution is to trap zero-impedance branch specification at the circuit data conversion process and replace zero impedances with appropriate small values.

--Rdugan 13:01, 3 September 2010 (UTC)
