

Avant!

Chapter 8

Using the .DC Statement

This chapter describes how to use the .DC statement to perform DC sweep analysis. It covers the following topics:

- [Understanding the DC Sweep Flow](#)
- [Performing DC Sweeps](#)
- [Using Other DC Analysis Statements](#)
- [Setting the DC Analysis Control Option](#)

Understanding the DC Sweep Flow

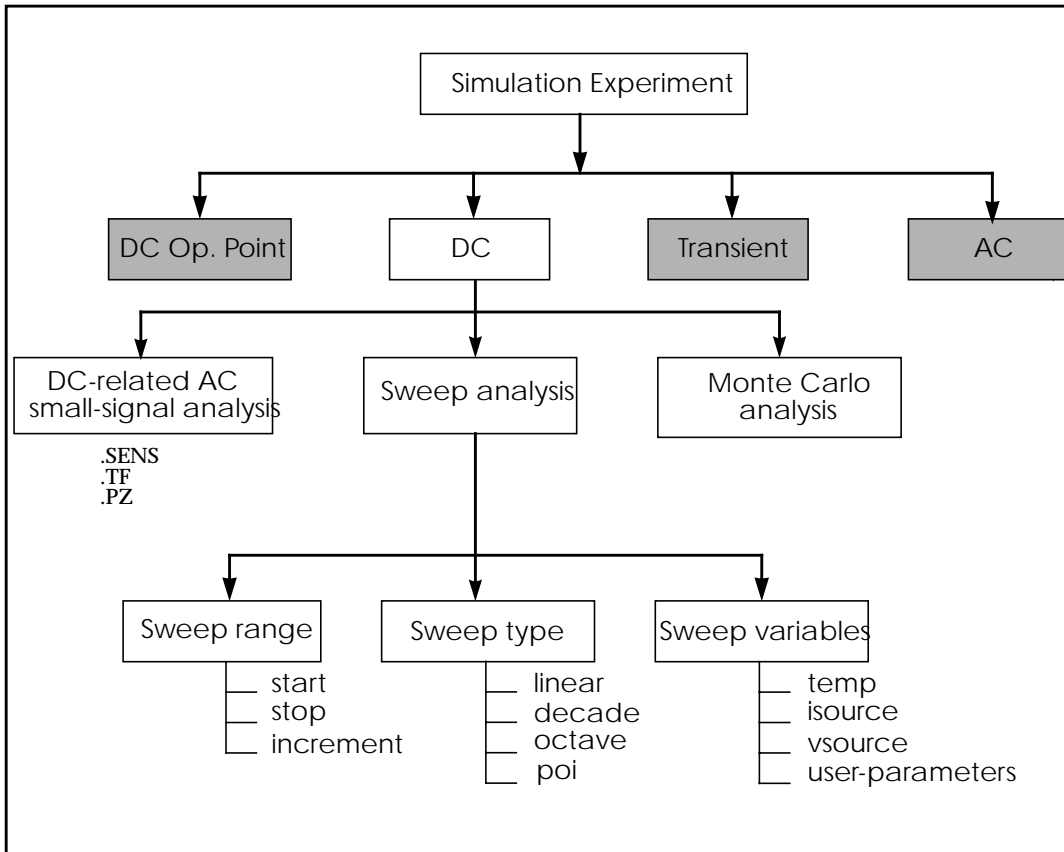


Figure 8-1: DC Sweep Overview

Performing DC Sweeps

The .DC statement is used in DC analysis to:

- Sweep any parameter value
- Sweep any source value
- Sweep temperature range
- Perform a DC Monte Carlo analysis (random sweep)
- Perform a DC circuit optimization
- Perform a DC model characterization

The format for the .DC statement depends on the application in which it is used, as shown in the following examples:

Syntax

Sweep or parameterized sweep:

```
.DC var1 START = start1 STOP = stop1 STEP = incr1
```

or

```
.DC var1 START=<param_expr1> STOP=<param_expr2>  
+ STEP=<param_expr3>
```

or

```
.DC var1 start1 stop1 incr1 <SWEEP var2 type np start2 stop2>
```

or

```
.DC var1 start1 stop1 incr1 <var2 start2 stop2 incr2 >
```

Data driven sweep:

```
.DC var1 type np start1 stop1 <SWEEP DATA=datanm>
```

or

```
.DC DATA=datanm<SWEEP var2 start2 stop2 incr2>
```

or

```
.DC DATA=datanm
```

Monte Carlo:

```
.DC var1 type np start1 stop1 <SWEEP MONTE=val>
```

or

```
.DC MONTE=val
```

Optimization:

```
.DC DATA=datanm OPTIMIZE=opt_par_fun
+ RESULTS=measnames MODEL=optmod
```

or

```
.DC var1 start1 stop1 SWEEP OPTIMIZE=OPTxxx
+ RESULTS=measname MODEL=optmod
```

The .DC statement keywords and parameters have the following descriptions:

<i>DATA=datanm</i>	datanm is the reference name of a .DATA statement.
<i>incr1 ...</i>	voltage, current, element, model parameters, or temperature increment values
<i>MODEL</i>	specifies the optimization reference name used in the .MODEL OPT statement used in an optimization analysis
<i>MONTE=val</i>	produces a number <i>val</i> of randomly generated values, which are used to select parameters from a distribution. The distribution can be <i>Gaussian</i> , <i>Uniform</i> , or <i>Random Limit</i> .
<i>np</i>	number of points per decade or per octave, or just number of points depending on the preceding keyword.
<i>OPTIMIZE</i>	specifies the parameter reference name used for optimization in the .PARAM statement
<i>RESULTS</i>	specifies the measure name used for optimization in the .MEASURE statement
<i>start1 ...</i>	starting voltage, current, element, model parameters, or temperature values Note: If type variation “POI” (list of points) is used, a list of parameter values, instead of “start stop” is specified.

<i>stop1 ...</i>	final voltage, current, any element, model parameter, or temperature values
<i>SWEEP</i>	keyword to indicate a second sweep has different type of variation (DEC, OCT, LIN, POI, DATA statement, or MONTE=val)
<i>TEMP</i>	keyword to indicate a temperature sweep
<i>type</i>	can be any of the following keywords: DEC — decade variation OCT — octave variation LIN — linear variation POI — list of points
<i>var1 ...</i>	name of an independent voltage or current source, any element or model parameter, or the keyword TEMP (indicating a temperature sweep). Star-Hspice supports source value sweep, referring to the source name (SPICE style). However, if parameter sweep, a .DATA statement, and temperature sweep are selected, a parameter name must be chosen for the source value and subsequently referred to in the .DC statement. The parameter name can not start with V or I.

Examples

The following example causes the value of the voltage source VIN to be swept from 0.25 volts to 5.0 volts in increments of 0.25 volts.

```
.DC VIN 0.25 5.0 0.25
```

The following example invokes a sweep of the drain to source voltage from 0 to 10 V in 0.5 V increments at VGS values of 0, 1, 2, 3, 4, and 5 V.

```
.DC VDS 0 10 0.5 VGS 0 5 1
```

The following example asks for a DC analysis of the circuit from -55°C to 125°C in 10°C increments.

```
.DC TEMP -55 125 10
```

As a result of the following script, a DC analysis is conducted at five temperatures: 0, 30, 50, 100 and 125°C.

```
.DC TEMP POI 5 0 30 50 100 125
```

In the following example, a DC analysis is performed on the circuit at each temperature value, which results from a linear temperature sweep from 25°C to 125°C (five points), sweeping a resistor value called xval from 1 k to 10 k in 0.5 k increments.

```
.DC xval 1k 10k .5k SWEEP TEMP LIN 5 25 125
```

The example below specifies a sweep of the value par1 from 1 k to 100 k by 10 points per decade.

```
.DC DATA=datanm SWEEP par1 DEC 10 1k 100k
```

The next example also requests a DC analysis at specified parameters in the .DATA statement referenced by the .DATA statement reference name datanm. Parameter par1 also is swept from 1k to 100k by 10 points per decade.

```
.DC par1 DEC 10 1k 100k SWEEP DATA=datanm
```

The final example invokes a DC sweep of the parameter par1 from 1k to 100k by 10 points per decade, using 30 randomly generated (Monte Carlo) values.

```
.DC par1 DEC 10 1k 100k SWEEP MONTE=30
```

Schmitt Trigger Example

```
*file: bjtschmt.sp      bipolar schmitt trigger
.options post=2
vcc 6 0 dc 12
vin 1 0 dc 0 pwl(0,0 2.5u,12 5u,0)
cb1 2 4 .1pf
rc1 6 2 1k
rc2 6 5 1k
rb1 2 4 5.6k
rb2 4 0 4.7k
re 3 0 .47k
*
diode 0 1 dmod
q1 2 1 3 bmod 1 ic=0,8
q2 5 4 3 bmod 1 ic=.5,0.2
*
.dc vin 0,12,.1
*
.model dmod d is=1e-15 rs=10
.model bmod npn is=1e-15 bf=80 tf=1n
+ cjc=2pf cje=1pf rc=50 rb=100 vaf=200
.plot v(1) v(5)
.graph dc model=schmittplot input=v(1) output=v(5) 4.0 5.0
.model schmittplot plot xscal=1 yscal=1 xmin=.5u xmax=1.2u
.end
```

Using Other DC Analysis Statements

Star-Hspice provides the following additional DC analysis statements. Each of these statements uses the DC equivalent model of the circuit for its analysis functions. For .PZ, capacitors and inductors are included in the equivalent circuit.

- .OP* specifies the time or times at which an operating point is to be calculated. See Chapter 5, “DC Initialization and Operating Point Analysis,” for information about DC operating point analysis.
- .PZ* performs pole/zero analysis (.OP specification is not required)
- .SENS* obtains the DC small-signal sensitivities of specified output variables with respect to circuit parameters (.OP specification is not required)
- .TF* calculates the DC small-signal value of a transfer function (the ratio of an output variable to an input source). An .OP specification is not required.

Star-Hspice also provides a set of DC control options and DC initialization statements that allow for the modeling of resistive parasitics and the initialization of nodes. These enhance the convergence properties and the accuracy of the simulation.

DC-Related Small-Signal Analyses

This section describes how to perform DC-related small signal analysis.

.SENS Statement — DC Sensitivity Analysis

If a .SENS statement is included in the input file, Star-Hspice determines the DC small-signal sensitivities of each specified output variable relative to every circuit parameter. The sensitivity measurement is the partial derivative of each output variable with respect to the value of a given circuit element, taken at the operating point and normalized to the total change in output magnitude. Therefore, the sum of the sensitivities of all elements is 100%. Sensitivities are calculated for resistors, voltage sources, current sources, diodes, and BJTs.

You can only perform one .SENS analysis per simulation. If more than one .SENS statement is present, only the last one is run.

Syntax

```
.SENS ov1 <ov2 ...>
```

ov1 ov2 ... branch currents or nodal voltage for DC component
 sensitivity analysis

Example

```
.SENS V(9) V(4,3) V(17) I(VCC)
```

Note: The .SENS statement can generate very large amounts of output for large circuits.

.TF Statement — DC Small-Signal Transfer Function Analysis

The transfer function statement (.TF) defines the small-signal output and input for DC small-signal analysis. When the .TF statement is included, Star-Hspice computes the DC small-signal value of the transfer function (output/input), input resistance, and output resistance.

Syntax

```
.TF ov srcnam
```

where:

ov small-signal output variable

srcnam small-signal input source

Examples

```
.TF V(5,3) VIN
```

```
.TF I(VLOAD) VIN
```

For the first example, Star-Hspice computes the ratio of V(5,3) to VIN, the small-signal input resistance at VIN, to the small-signal output resistance measured across nodes 5 and 3. Only one .TF statement can be used per simulation. If more than one .TF statement is present, only the last is performed.

.PZ Statement— Pole/Zero Analysis**Syntax**

```
.PZ ov srcnam
```

where

ov output variable: a node voltage V(n), or branch current I(element)

srcnam input source: an independent voltage or current source name

Examples

```
.PZ V(10) VIN
```

```
.PZ I(RL) ISORC
```

See Chapter 23, “Pole/Zero Analysis,” for complete information about pole/zero analysis.

Pole/Zero Analysis Options

Control options are set using the .OPTIONS statement. The following are the control options used in pole/zero analysis.

<i>CSCAL</i>	sets the capacitance scale. Capacitances are multiplied by CSCAL. Default=1e+12.
<i>FMAX</i>	sets the limit for maximum pole and zero frequency value. Default=1.0e+12 · FSCAL.
<i>FSCAL</i>	sets the frequency scale. Frequency is multiplied by FSCAL. Default=1e-9.
<i>GSCAL</i>	sets the conductance scale. Conductances are multiplied by GSCAL. Resistances are divided by GSCAL. Default=1e+3.
<i>ITLPZ</i>	sets the pole/zero analysis iteration limit. Default=100.
<i>LSCAL</i>	sets inductance scale. Inductances are multiplied by LSCAL. Default=1e+6.

The scale factors must satisfy the following relations:

$$GSCA = CSCAL \cdot FSCAL$$

$$GSCAL = \frac{1}{LSCAL} \cdot FSCAL$$

If scale factors are changed, it might be necessary to modify the initial Muller points (X0R, X0I), (X1R, X1I) and (X2R, X2I), even though Star-Hspice multiplies initial values by (1e-9/GSCAL).

PZABS sets absolute tolerances for poles and zeros. This option affects the low frequency poles or zeros. It is used as follows:

If $(|X_{real}| + |X_{imag}| < PZABS)$,

then $X_{real} = 0$ and $X_{imag} = 0$.

It is also used for convergence tests. Default=1e-2.

PZTOL sets the relative error tolerance for poles or zeros. Default=1.0e-6.

RITOL sets the minimum ratio value for (real/imaginary) or (imaginary/real) parts of the poles or zeros. RITOL is used as follows:

If $|X_{imag}| \leq RITOL \cdot |X_{real}|$, then $X_{imag} = 0$

If $|X_{real}| \leq RITOL \cdot |X_{imag}|$, then $X_{real} = 0$

Default=1.0e-6.

(X0R,X0I), the three complex starting points in the Muller pole/zero analysis algorithm are:

(X1R,X1I), X0R = -1.23456e6 X0I = 0.0

(X2R,X2I), X1R = -1.23456e5 X1I = 0.0
X2R = +.23456e6 X2I = 0.0

These initial points are multiplied by FSCAL.

Setting the DC Analysis Control Option

The following control option is used in DC small signal analysis.

ITL2=val sets the DC transfer curve iteration limit. Increasing the iteration limit can be effective in improving convergence only on very large circuits. Default=50.