

# Intusoft Newsletter

Personal Computer Circuit & System Design Tools



Copyright © Intusoft, All Rights Reserved

Issue #74 June 2004  
Tel. (310) 329-3295  
Fax (310) 329-9864

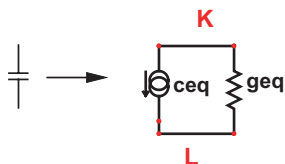
## IC Stepping - A New and Unique DC Convergence Option

### Problem:

DC Convergence in SPICE simulators cannot be achieved for analog feedback circuits where high gain amplifiers are cascaded and with active regions that are offset. These circuits work perfectly in practice, however, today's standard SPICE convergence algorithms tend to oscillate and find a solution only by lucky chance. This is due to the chaotic nature of the numerical iteration process. In practice, such circuits do not oscillate in the time domain because capacitors are used to dampen the oscillation in accordance with control system theory.

### Solution:

A new convergence stepping algorithm was implemented to solve this problem. In this algorithm, each capacitor that has an Initial Condition (IC)=xxx will have a conductor in parallel with a current source placed across it. The current,  $ceq$  is the product of the IC voltage and the conductance,  $geq$ . The conductance starts off very high and is stepped down toward zero for each successful iteration. At the final iteration,  $ceq$  and  $geq$  are set to zero. It's probably easier to visualize it using a Thevenin Equivalent consisting of a resistor in series with a voltage source. But the Norton version slips directly into the SPICE admittance matrix as shown below.



$geq$	$-geq$	K row	$ceq$
$-geq$	$geq$	L row	$ceq$
K col	L col		

Figure 1: The SPICE matrix stamp for a capacitor.

### In This Issue

- 1 IC Stepping - A New and Unique DC Convergence Option
- 3 Stepping into a black hole (Timestep too small)
- 5 New Accounting Output
- 6 Power Deluxe gets new PwmPrimitive library
- 7 ExtractMid and ZapDelay script
- 8 Modeling PFCs
- 12 Footprint and Part Number pickers
- 14 Decipher SPICE2 Poly functions
- 15 New IBIS to SPICE converter
- 15 Intusoft's New Modeling Service

The following new IsSpice4 options control the new procedure:

**.OPTION ICSTEP = [num]**  
**.OPTION GEQFREQ = [omega]**

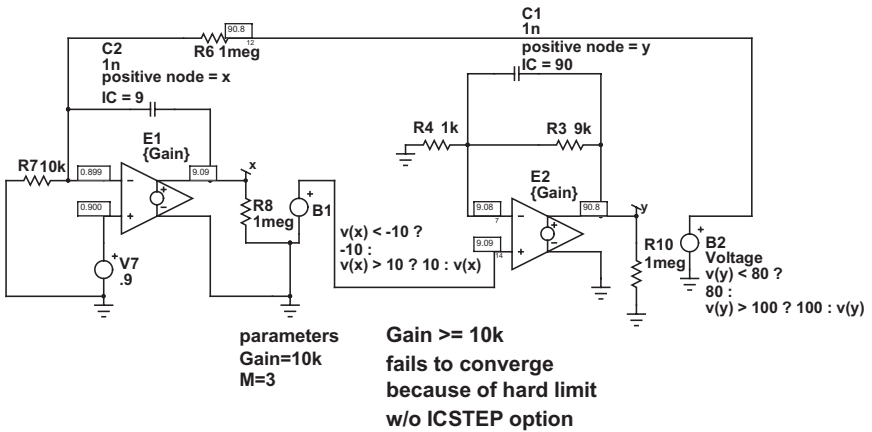
where num is a value between 1 and 100 that sets the number of steps and omega is an effective radian frequency, it defaults to 1e12 if you don't set it.

The initial geq is set to the product of GEOFREQ \* CapValue and ceq = geq\*InitCapVoltage.

**Discussion:**

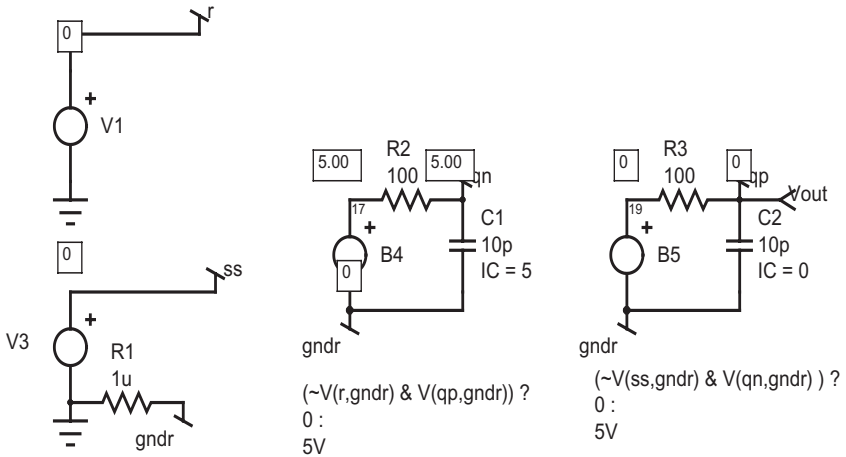
By way of example, for a 1nF capacitor, the shunt conductor would start at 1000 Siemens and finish at 1e-9 before finally being zeroed in the final step. It's clear that for cascaded op-amp circuits that the initial geq, ceq circuit will initialize the circuit properly because the high gain op-amps are forced to nearly the correct answer by their negative feedback. It may be that only one step is needed (ICSTEP=1) if the IC voltage is chosen correctly. This option will also correctly initialize digital behavioral circuits for DCOP and AC analysis.

Here's an example taken from the collateral material supplied in Newsletter 72.



**Figure 2:** With ICSTEP=1, this circuit converges in 6 iterations, without the option convergence fails after 445 trials.

Not only does the circuit converge with a gain of 10k, you can even increase the gain to 10e10 and get convergence using the ICSTEP option. Next is a behavioral flip-flop that's commonly used in Switch Mode Power supply control logic.



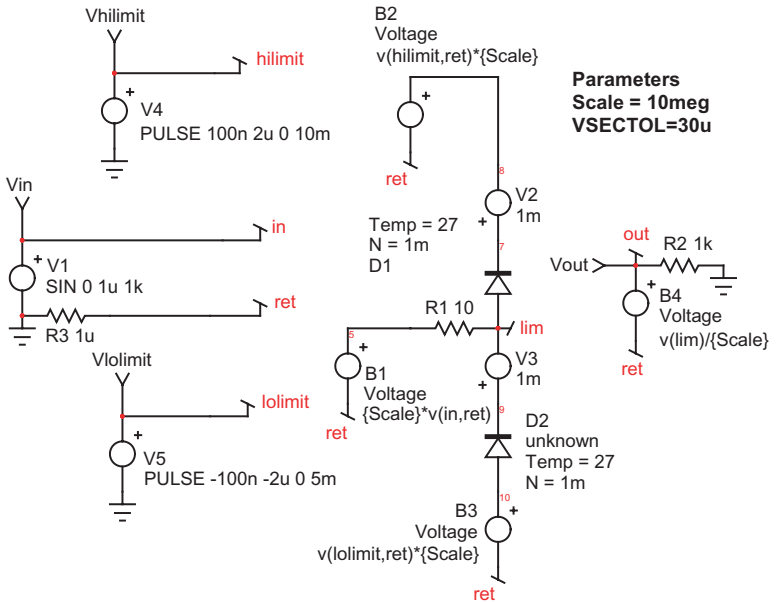
**Figure 3:** Converging a bistable circuit with ICSTEP=1

With this new capability, you can retain bistable models in your AC analysis and use capacitor ICs to initialize the circuit state.

## Stepping into a black hole (Timestep too small)

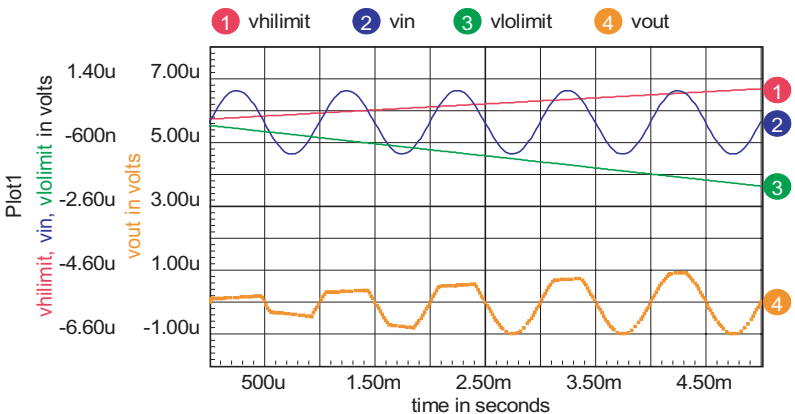
SPICE can encounter singularities, not unlike a cosmological black hole. Except SPICE doesn't have any of Hawking's quantum magic to escape the singularity, at least not yet. There is an implicit assumption by SPICE simulators that reduction in time step will remove singularities. Clearly this happens if R-C networks are in the signal flow. However, the use of behavioral elements can confound that logic. Intusoft's [VSECTOL](#) can exacerbate the phenomenon; consider a behavioral divider whose denominator sweeps through zero. The usual trick of adding a small constant to the denominator will fail because VSECTOL will draw the solution right into the singularity. We typically avoid this problem by testing the denominator using an [if-then-else](#) expression before evaluating the expression; for example to divide A by B, the expression would be:  $v=B > 0 ? A/(B+1u) : LIMITVAL$ , where LIMITVAL is something reasonable. Recall that the SPICE numerical iterations can produce intermediate results with B near 0 so that this technique works if B never has a final solution near 0. But if it does, then the jump to LIMITVAL can cause a singularity that VSECTOL can be drawn into. Just passing the result through an R-C network doesn't exempt the singular node. So what's needed is a limiter configuration that doesn't have a singularity. To accomplish this we used diode limiters that have been scaled to limit around 1 to 100 volts so we can use a fixed emission coefficient (See [NL73](#)).

Here's what the new limiter looks like:



**Figure 4:** Limiter without singularities

Now we can use this circuit to limit B to prevent a divide by zero. The value of LIMITVAL is of no consequence since the divide singularity is not encountered in the final solution. Figure 5 illustrates how this works using signals that represent switching times with variable limits. VSECTOL controls the simulation step using the internal scaled values.



**Figure 5:** Diode Limiters prevent timestep too small errors.

# New Accounting Output

The account output that is tacked onto the end of the .OUT file has been changed for the next software release. In SpiceNet go to Actions > Simulation Setup > Edit, click on the "Simulator Options..." button and check the "ACCT Circuit Accounting Information" box to enable this option. This .options acct flag is used to produce a summary listing of simulation related information, such as number of iterations for various operations and simulation time for various analyses.

Two new lines have been added to this list. The first line reports the Maximum Number of Transient Iterations that were needed in the simulation. The next line reports the simulation time step at which this number of iterations occurred. By setting ITL4 option to a large number, typically 500, the user can get an indication of how hard the simulator has to work for a particular transient simulation setup, and thus adjust the circuit and simulator options accordingly for the most accurate result and/or best simulation run time.

Let's consider the following power supply design example. This ROYERTST.DWG is included in the Circuits/Power folder of our Power Deluxe packages. First run the simulation with only ITL4=500, Integration Method=TRAP and ACCT options set.

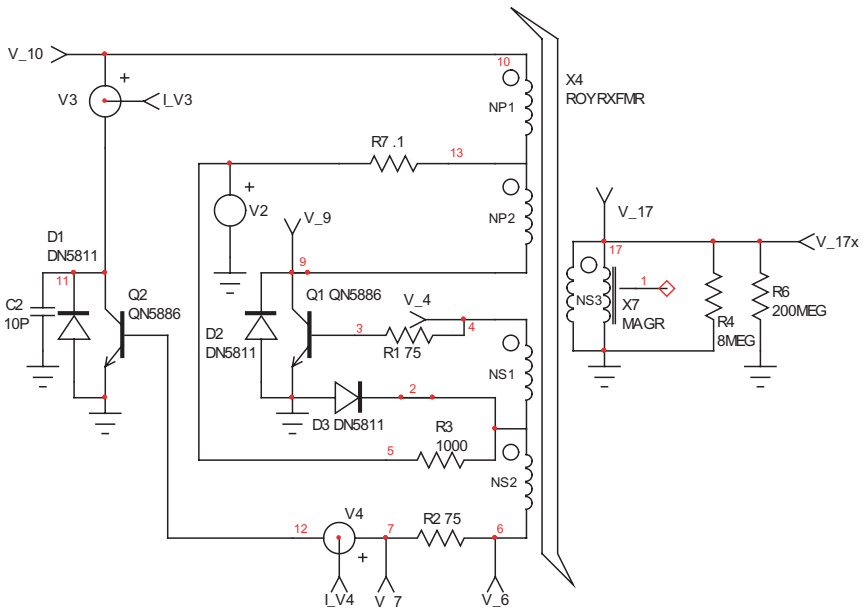


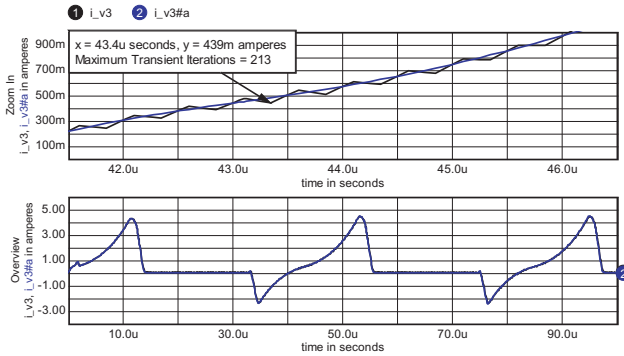
Figure 6: Example Power Supply Design circuit.

Notice the noise-like oscillation in the output current I\_V3. The following 2 lines from the output file indicate that the simulator had to work hard at 43Us to get convergence.

Maximum Transient Iterations = 213

Maximum Transient Iterations Time = 4.3353E-005

This corresponds to the area where we noticed the noise in I\_V3. Since this kind of noise is an attribute of numerical integration methods, we will run the simulation again, but this time selecting Gear as the integration method in the simulation options. Notice that the noise is now eliminated and the maximum number of iterations is reduced to 20!



**Figure 7:** The black waveform is  $i_{v3}$  using trapezoid integration (noise-like oscillations) and the blue waveform is  $i_{v3}$  using gear integration (smooth curve).

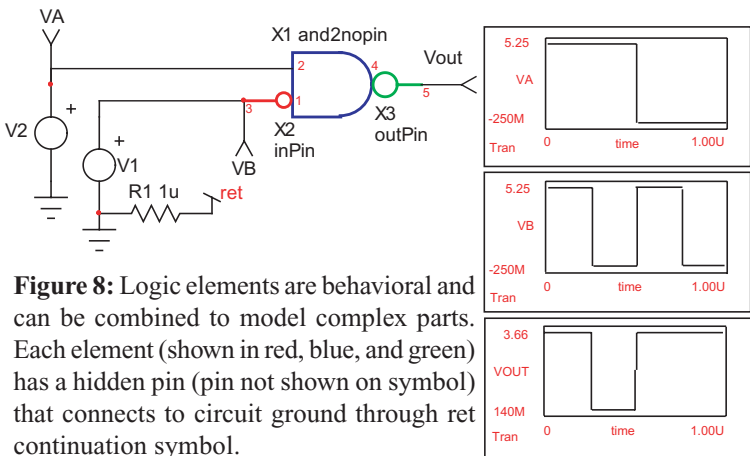
## Power Deluxe gets new PwmPrimitive library

The test circuit in Figure 4 contains a second configuration with only the model. The model was saved as a subdrawing, and using the Library Manager, exported to a library file. This model and many others are being included in Intusoft's Power Deluxe libraries in a new library called PwmPrimitive. The library consists of a set of models frequently used in Switch Mode Power Supplies. Each of the models uses an isolated ground so that the resulting circuit can have substantial offsets from SPICE ground. The logic elements are behavioral and the gates have shortened pins or no pins at all. This was done to allow input and output inverters to be attached to the gates as if they were pins. Figure 8 illustrates how this works along with the resulting waveforms. Each of the models has a hidden pin that connects externally to the circuit ground. The circuit ground can be on the primary side of a SMPS; it doesn't have to be at SPICE ground.

Here's what's in the new library as of now:

- OR Gates: 2,3,4 pins
- AND Gates 2,3,4,5 pins
- Input inverter pin.
- Output inverter pin
- D flip flop
- R-S flip flop
- Comparator with hysteresis
- Limiter, Scaled described above
- UVLO generic under voltage lockout
- Oscillator, generic RCT oscillator
- Constant power load
- MOS gate capacitor load for testing drivers
- Opamp3, generic with parameters for output limits, etc.
- Totem pole MOS Driver
- BiCMOS MOS Driver
- PFC multiplier

We've used this library to model the MC33260 PFC switching and average models and we will be adding more SMPS models and primitive models.



**Figure 8:** Logic elements are behavioral and can be combined to model complex parts. Each element (shown in red, blue, and green) has a hidden pin (pin not shown on symbol) that connects to circuit ground through ret continuation symbol.

## ExtractMid and ZapDelay script

“ExtractMid” and “zapDelay” IntuScope scripts can be used to get rid of start-up transients before doing an operation such as an FFT. These scripts take advantage of ICAPS’ ability to access specific elements in a vector array.

To access a specific element you use `vec[n]` to access the n'th element of vector `vec`. To access a range of elements you use `vec[(m,n)]` to access all elements between the m'th to n'th element of vector `vec`. Note that if n is less than m then the order of elements is reversed in

the result.

The problem with the above method is that it requires you to know what index corresponds with what value. You can obtain this information easily by typing "print vec" in the Commands window, then Actions > Do Command Script (Ctrl+R) to display all vector elements of vec in the Output window and their corresponding index. An easier way to extract a mid range of vector values is to extract data based on the default scale vector range. If for example we create a new vector based on data between 50n to 75n, we would use `vec[[50n,75n]]`. The ExtractMid script brings up a dialog that allows you to specify the default scale range you want to extract and plots the new vector. The zapDelay script subtracts the offset so your new vector starts at 0.

## Modeling PFCs

Power Factor Correction (PFC) circuits are used to smooth the current waveforms in power mains by making the real power ( $\text{average}[I(t)*V(t)]$ ) equal to the apparent power ( $I_{rms}*V_{rms}$ ). There are 2 benefits of using PFCs. First, the distribution losses are minimized. Minimizing distribution losses lets you get the most out of you infrastructure — for example, running more equipment on the same circuit breaker. Second, power line harmonics are minimized. Minimizing harmonics makes a cleaner environment for other circuits. Sensitive equipment, such as microphones and medical instrumentation, are the main beneficiaries of a low-noise environment. PFC's use switch mode techniques to make the load seen at the power mains appear resistive. This is accomplished by chopping the input current waveform into a series of pulses, applying a control law, then reconstituting the waveform as a DC output (i.e., by using a capacitor to hold up the output voltage when the instantaneous input power is less than the load power). All commercially available PFCs try to match the input current waveform to the rectified main's voltage. They generally use a boost topology, although flyback is sometimes used if a secondary regulator is not required. The bandwidth of a PFC must be lower than the rectified main's frequency. For commercial power systems around the world, that places the bandwidth around 10Hz. Switching frequencies are set by economics. As switch frequencies are increased, the size and cost of the magnetic components decrease until second order effects begin to dominate. Similarly, the semiconductor switches will have increased losses at higher frequencies. Capacitors and interconnections will deteriorate in quality at higher frequencies. Using today's technology, frequencies run from 50kHz up to several hundred kHz. There is a gradual trend toward higher frequencies as



critical component behavior is improved — for example, eliminating diode reverse recovery transients.

The switch level model of a PFC using a SPICE simulator is pretty straightforward. You simply interconnect the parts the way the IC manufacturer depicts them in their data sheet. Use behavioral models for the logic and component-based models for the semiconductors. Magnetic device models can be extracted using magnetic device design programs such as Intusoft's Magnetics Designer. These models take on the order of a minute to simulate 1 millisecond of operation. Only several milliseconds need to be simulated for DC-DC converters. But, a PFC may need to be observed for several seconds to view its start-up characteristics. Trying that with a switching model leads to simulations that may take days to run and require more RAM than the current microprocessors support.

Another type of model is an "Average" model. Average models replace the switches with the average behavior over one or more switching cycles. Average models in general run 100 times faster than switch-based models. One of the more straightforward mathematical implementations is to use difference equations and unit time delays; that is, Z-Transforms. Z-Transforms have been around a long time and are used to describe sample data systems. In theory, they work up to  $\frac{1}{2}$  the switching frequency. The difference equations accurately describe the system only at the switching instants. They will describe the switched inductor current only at the switching instants. The average value must be computed based on the behavior of the current over the entire switching cycle. These average currents are used to model the input and output current, while internally the model works at the switching instants. With a Z-Transform based model, it's possible to get a DC operating point and to describe the AC transfer function. The AC analysis falls out because the unit time delay is a simple transmission line that has well-defined frequency domain characteristics.

PFC average models are somewhat different than the ones used for DC-DC converters because they must account for input voltage approaching zero. At some point, the converter will not be able to generate enough flyback voltage to produce any output. This is a unique mode and can be a source of line harmonics, especially for critical mode or discontinuous mode controllers (See Mode Sidebar).

Intusoft is in the process of modeling the 3 basic PFC topologies: Critical Mode, CCM and DCM. The grandfather of PFC's, the UC1854 (otherwise known as UC3845), is usually used as a CCM device. Building the switched model and using the Unitrode reference design, you can observe (Figure 9) that it never drops into DCM. Instead, a new mode is entered when the charging current is too small

to generate the required voltage for the output to conduct. When that occurs, the initial current at the beginning of the switching cycle is negative. For all input voltages greater than zero, there is an average input current. The following equations describe what happens:

F = switching frequency

T = 1/F

Don = Commanded duty ratio, Ton/T

Dr = Duty ratio of rise time, Tr/T

Doff=Duty ratio for output conduction, (1-Don-Dr) or zero

L = switched inductor value

C = switch capacitance

Tr = voltage rise time

During Tr, the average output voltage is  $V_p * 2/\pi$  and we assume the current change will be  $(V_p * 2/\pi - V_i) * Tr/L$ .  $V_p$  is mode dependant. If the diode conducts, it's the output voltage for a boost mode controller. Otherwise it's the peak flyback voltage. Rather than going through the detailed equations (you can find them in the hierarchical model posted on the web for our users) we will point out some of the tricks and pitfalls to be aware of when making these kinds of models.

One of the more annoying problems is the tendency for the numerical iterations to divide by zero. You must protect divide operations as we have shown in the past as follows:

B node1 node2  $v=v(\text{den}) > 0 ? v(\text{num})/(v(\text{den}) + 1u) : v(\text{num})/1u$

It's usually pretty clear that there isn't a useful solution for  $v(\text{num}) == 0$ , however, during the process of numerical iteration the zero value is likely to occur, usually on the first iteration. When that occurs, SPICE3 places BIGNUM in the result. BIGNUM is  $1e100$ . Such a large number makes the matrix nonsense and you get a convergence failure. Notice that we also match a smooth transition in the else case so the solution can get back on track.

If you use transcendental functions such as `asin()`, then hidden in the evaluation of its derivative is a divide by zero. Remember derivatives are used to load the small signal conductance back into the matrix. You should use a polynomial approximation or piece-wise linear table to get rid of that one. We've added a warning in our next update if this type of operation occurs. When you see this new warning, you may also get seemingly unrelated warnings on other devices. That's because BIGNUM messed up everything!

It seems that no matter how careful you are, mode switches will add a glitch into your model. It's common to oscillate or get convergence failures near mode switch events. There are 2 ways to deal with the problem. First, introduce hysteresis in the mode switching logic; you can use the SPICE3 switch to do that. Second, make a fuzzy-smooth

transition. Borrowing from fuzzy logic, you can join 2 control modes using a sigmoid function. The sigmoid function:

$$V_s = 1 / (1 + \exp(-M * V_i))$$

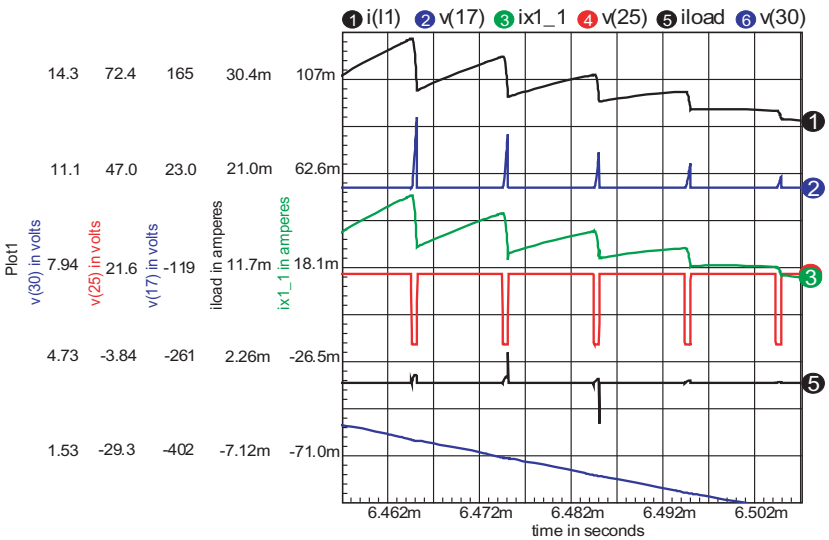
Goes between 0 and 1 as  $V_i$  passes through zero.  $M$  determines the direction and smoothness of the transition. If  $V_i$  is used for mode detection, you can join to computation, say for Doff, as follows:

$$\text{Boff Doff } 0 \quad v = v(V_s) * v(\text{mode1}) + (1 - v(V_s)) * v(\text{mode2})$$

You can adjust  $M$  to the degree of smoothness needed. This model is now available as fuzzyMux on our web site.

### Mode SideBar

Switch mode power supplies generally use a diode as one of the switches. When either the diode or the semiconductor switch is always conducting, the mode is called continuous conduction mode, CCM. CCM operation is easy to model because you can use superposition. Discontinuous conduction mode (DCM) occurs when neither the diode or the switch conduct. DCM is more difficult to model because you must break the operation into 2 parts - when the switched inductor conducts current and when it doesn't. This is a fairly simple view because we've failed to account for resonance or forced commutation. Still, these models perform well in the right application. A third mode called Critical Mode occurs when the converter is always in the transition between CCM and DCM. By operating at a constant peak input current, the frequency is varied to modulate the power delivered.



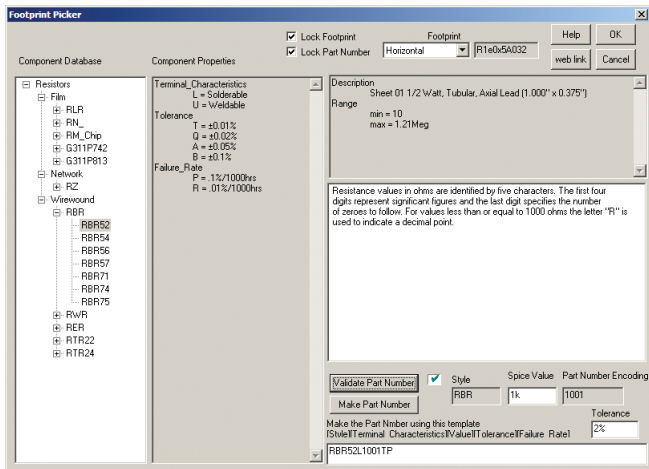
**Figure 9:** The input current continues to conduct even when the load current vanishes. Waveforms are: 1) input current 2) flyback voltage 3) switch current 4) output current 5) rectified input

## Footprint and Part Number pickers

NASA maintains the NASA Parts Selection List, NPSL for electronic parts. <http://nepp.nasa.gov/npsl/index.htm>. This list consolidates all of the military and space electronic parts. Of particular interest is the comprehensive list of passive parts (resistors, capacitors and inductors).

If you were to flatten out all of these part numbers, the count would be in the millions, just for the RN style metal film resistors [1]. Adding that to the SpiceNet browser part database would cause a severe performance penalty. Instead, Intusoft has made an XML database [2] for each generic Spice part called RR.type for resistors, CC.type for capacitors and LL.type for inductors. When these files are present in the library folder, the XML file is parsed into a “Footprint picker” dialog. The purpose of this dialog is to select an appropriate footprint to be used by the Bartels layout program and to generate a unique part number that can be used in a Bill of Materials to purchase the parts. Users can add to or modify the xml files and even write DLL’s to build part numbers. The xml database provides sufficient information for the user to construct a proper part number and a selection of legal footprints. Intusoft has implemented the ability to build and verify the NPSL part numbers because part number construction is error prone. Commercial equivalents should be readily available, although the user will probably have to substitute the specific vendor part number instead of the NPSL part number.

To convert a Spice R L or C into a specific part, right click on the symbol and select “Footprint...” from the dropdown menu. Navigate through the tree to select the desired part and press the “Make Part Number” button and choose the various attributes from the wizard.



When you press OK, the part number will be entered into the SpiceNet part number field and the mechanical package footprint will go into the “Type” field. Both the “Part Number” and “Type” will be converted to read only so that you must use the “Footprint picker” to make changes.

For capacitors, the rules for constructing part numbers vary by part type. Many part numbers are selected from tables that assign a dash number to encode various properties. Intusoft has replicated these tables in “.RTF” files in the library folder. When a <type>.RTF file exists, the table will be loaded into Footprint Picker dialog. Clicking in the appropriate dash number field, followed by pressing the Make Part Number button, will encode the selected dash number. A Capacitor wizard dialog will then be displayed for you to define additional properties. When the selected item has an associated part size then the correct footprint will be automatically selected. Footprints, Tolerances and Part Numbers will all be properly set the next time you bring up the dialog. (Ongoing work to be done includes filling out the correct radio buttons in the Wizards).

For our defaults, resistor pins and capacitor pins using the standard symbol are assumed swappable. Capacitor pins using the c.2 or c.3 symbols are assumed not swappable. You can change these defaults using the Mechanical Properties dialog/Footprint Prop. button.

[1] 1409 0.5% parts \* 3 lead characteristics \* 6 temp coefs \* 6 tol's \* 4 reliabilities \* 6 sizes = 3.65 million variations.

[2] A portion of the RR.type xml database

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- a browser database for resistors
      Footprints should be name=value comma separated
      like Horizontal Mount=H301 -->
<Resistors>
  <Film>
    <Link href="http://nepp.nasa.gov/npsl/Resistors/film_type.htm"> </Link>
    <RLR>
      <PartNumber val="RLR[size]C[Value][Tolerance][Failure Rate]">
      </PartNumber>
      <Temperature Coefficient="±100ppm/°C" Max_No_Load="150C"
      Max_Load="70C"> </Temperature>
      <Tolerance F="1%" G="2%" EndOfLife="4%"> </Tolerance>
      <Reliability P=".1%/1000hrs" R=".01%/1000hrs" S=".001%/1000hrs">
      </Reliability>
      <RLR05>
        <Description>
          1/8 Watt, Tubular, Axial Lead (0.150" x 0.066")
        </Description>
        <Range min="4.7" max="1Meg"> </Range>
        <Footprint Horizontal="R150x066A016" Vertical="R02P45B10">
        </Footprint>
      </RLR05>
    </RLR>
  </Film>
</Resistors>
```

## Decipher SPICE2 Poly functions

SPICE2 allows circuits to contain dependent sources characterized by any of the following four equations  $i=f(v)$ ,  $v=f(v)$ ,  $i=f(i)$ , or  $v=f(i)$ . The functions must be polynomials specified by a set of coefficients and the arguments may be multidimensional. Both the number of coefficients and the number of dimensions are totally arbitrary. The meaning of the coefficients depends upon the dimension of the polynomial.

SPICE3 added the B element to enable you to use any arbitrary equation. With the B element, you are no longer limited to only polynomial functions. To maintain backward compatibility with SPICE2 based software, Intusoft automatically converts the hard to decipher poly syntax to an equation. Viewing the converted syntax makes it much easier to figure out what is going on.

Normally, only the user-generated netlist is placed in the output file. The actual simulated netlist, however, is in a slightly different format. To place the simulated IsSpice4 netlist in the output file you need to enable the LIST option. This is done by selecting the IsSpice4 Simulation Setup, pressing the "Simulator Options..." button, "More..." button, and Screen/File Tab, then check the "LIST Output the complete netlist" box. This netlist has all subcircuits flattened, and any SPICE2 polynomial syntax for the E, F, G, or H elements are translated into the B element syntax.

Another method to invoke this polynomial syntax conversion is to use ICAP/4's ICL listing command. After a simulation, the Simulation Control dialog appears. Type in the Simulation Control Script Window "listing deck" and press the DoScript button. The Output Window will contain the netlist.

### Generate four types of netlists with listing command.

listing deck- Converted netlist read in by the simulator.

listing physical-Converted netlist with line numbers.

listing logical- Converted netlist with line numbers and continuation lines collapsed into one line. Hides lines starting with "\*" (comment).

listing expand- Converted netlist with line numbers and subcircuits *expanded*. Uses node:subckt\_ref syntax. Remove :x# in equations. Makes it easy to see if node is an internal or external subckt node.

### User Generated Netlist

```
EOS 3 1 POLY(1) 31 98 1E-3 1
EREF 98 0 POLY(2) 99 0 50 0 0 0.5 0.5
ECM1 30 98 POLY(2) 1 98 2 98 0 0.5 0.5
GSY1 99 52 POLY(1) 99 50 4E-3 -2.6E-4
GSY2 52 50 POLY(1) 99 50 3.7E-3 -.6E-3
```

### "listing deck" ICL command

```
b_eos 3 1 v = 1e-3 + v(31,98)
b_eref 98 0 v = 0.5 * v(99) + 0.5 * v(50)
b_ecm1 30 98 v = 0.5 * v(1,98) + 0.5 * v(2,98)
b_gsy1 99 52 i = 4e-3 + -2.6e-4 * v(99,50)
b_gsy2 52 50 i = 3.7e-3 + -.6e-3 * v(99,50)
```

## New IBIS to SPICE converter

IBIS is the **I**nput/**O**utput **B**uffer **I**nformation **S**pecification established by the Electronics Industry Alliance. It is a modeling technique that provides a simple table-based buffer model for semiconductor devices. The IBIS models can be used to characterize I/V output curves, rising/falling transition waveforms, and package parasitic information for the device. IBIS models are intended to provide nonproprietary information about I/O buffers. Historically, the worry has been that SPICE netlists expose circuit designs and model parameters, which are considered intellectual property. IBIS, however, only exposes tables of data for device I/O pins.

Intusoft has created a new IBIS2SPICE converter that converts IBIS models through version 3.2. All you need to do is specify which IBIS file to convert and a SPICE netlist is outputted. This new program is free of charge with the upcoming 8x11 release for the following packages: Test Designer, ICAP/4 Professional, ICAP/4Windows Power Deluxe and ICAP/4Windows RF Deluxe.

## Intusoft's New Modeling Service

Most custom model development is charged at an industry rate of \$1,000 - \$20,000 per model depending on complexity. However, Intusoft felt that as a continuum to its outreach program to the engineering community, it would provide a FREE custom modeling service. It is targeted for Intusoft customers and the SPICE community at large, and excludes IC manufacturers from utilizing the service.

The service starts by an inquirer filling out a Model Request Form (MRF) from the "Free Modeling Service" link on Intusoft's home page. A reply is furnished from Intusoft within 48 hours of a request. Another reply follows on, which informs the requester of modeling feasibility, approximate time to completion, and the possibility of a charge, i.e., for very sophisticated parts requiring many hours of design and verification time.

To date, Intusoft's modeling team has completed more than 150 parts from the service as indicated on the respective link from our home page. The link includes monthly updates of completed models from engineers who have requested our expertise in developing SPICE models. This is an unbelievable and unheard of service so take advantage of it if you have component models you would like to request from us. Also check out our growing list of new models for the upcoming ICAP/4 8x11 release this summer.

Check it out at: <http://www.intusoft.com/models/NewModelsUpdate.htm>

**Sample list of requested models:** MC33260D (PFC), TLV2370 (TI OpAmp), AQV414S (PhotoMOS Relay), DG9411DL (Low Voltage Single SPDT Analog Switch), AD8611ARM (Ultrafast 4n Single Supply Comparator), LMH6718 (High speed gain buffer), UC3842 (PWM controller), STB12BK80ZT4 (MOSFET), DVR135 (TI Line driver)