AC 2007-2333: ANALOG INSYDES — A NEW TOOL FOR TEACHING INTRODUCTORY CIRCUITS

Jean-Claude Thomassian, Georgia Southern University

Dr. Jean-Claude Thomassian received his BS degrees in Electrical Engineering and Mechanical Engineering from the University of Toledo in 1992 and 1993, respectively, and MS and Ph.D. degrees in Electrical Engineering from The University of Toledo in 1995 and 2002. His main professional interests are in mixed mode IC design and electrical engineering education; his recent research activity concentrates on symbolic analysis of circuits and MOS models.

Analog Insydes—A New Tool for Teaching Introductory Circuits

Abstract

In this paper, Analog Insydes is used to deepen the student understanding of introductory circuits and to serve as a design tool for circuit analysis. Modern four-year electrical and computer engineering baccalaureate engineering programs are crowded. As a consequence, too little time is made available to teach introductory electric circuits in the traditional manner. The best way to improve the outcome of what can be accomplished in the shortest possible technique, given the time constraint, is to base it strictly on the newly developed symbolic circuit simulators such as Analog Insydes. Three examples of applications are presented to show the proposed method at work, a simple DC parallel resistor circuit, a current controlled current source, and an RC low-pass filter. A concluding section offering the pedagogical potential of the package is assessed. Also some remarks regarding the challenges of integrating it into an undergraduate program of instruction are offered.

Introduction

Analog Insydes 2.1 (AI) package is now on the market¹. It is distributed by Wolfram Research, Inc. as a toolbox of Mathematica². Essentially, it is a symbolic circuit analysis package, with several unique features: It is designed to be used interactively with any of the SPICE³ family; it too has a limited device model library, and it is able to import and read SPICE files. It is basically constrained to solving linear circuits using the Laplace Transform; these may be either linear circuits or linearized small-signal electronic circuits. Analog Insydes uses some recent advances in approximation techniques; the addition of these methods and practical features distinguishes it from general purpose computer algebra systems (CAS).

The more or less conventional methods of teaching introductory electric circuits are not satisfactory; this paper proposes an alternative we believe to have the potential to achieve superior results^{4,5}.

Computer Circuit Analysis

The familiar use of numerical circuit simulators such as PSpice⁵ calls for no particular elaboration. The new development of symbolic circuit simulators on the other hand is much less well known¹. The purpose of this paper is to bring attention to them and their potential for use in teaching introductory courses. It may be useful to remind our readers of the difference between a simulator and a general purpose equation solver: the simulator employs built-in routines for setting up the necessary set of application-specific equations to be solved, beginning from an easier to visualize problem description—in our case, the circuit schematic; then it uses the general purpose solver in the simulation. It is precisely the elimination of this tedious and error prone step plus the shortcut of using pre-existing model libraries that makes simulators so economical to use. It is precisely this vastly improved use of time that we advocate bringing to the teaching and learning of introductory circuits. Pedagogically speaking, pure numerical circuit

simulators, powerful and indispensable as they are, are not a full plate. They are not well suited to identifying cause and effect relationships nor do they lend themselves readily to isolating and ranking multiple contributions to a result. It has been claimed and we agree that “simple formulae are the salt of understanding.” How then do we get the computer to give us simple formulae, especially when the problem to hand may not be simple?

With almost slight exceptions, symbolic methods are restricted to linear or approximated- as-linear systems. Even simple circuits typically give rise to symbolic solution expressions of formidable complexity; moderate examples can yield answers requiring dozens or even hundreds of lines to print out. Such results are essentially worthless; the resulting impasse has been called “the complexity of expression barrier.” It is impossible to simplify or approximate algebraic expressions without some form of guidance as to the relative importance of the terms. Such can originate via experience and intuition or more formally by use of a so-called reference or model circuit. A model is simply the example under investigation, complete with a set of representative numerical values. Before dismissing the idea, one should be aware of two important facts: the numerical values or design point can be changed at the pleasure of the user, and more important still, the computer itself can now do simplification. A 200-term expression can be reduced to a two or three term expression by the push of a key. It is elementary to obtain a progressively simpler sequence of approximations with known bounds on the error, i.e., the difference in values yielded by the exact and the simplified expressions.

We proceed next by three examples of symbolic computer solution typical of what would be done in beginning courses. In a last section we describe why we believe introductory circuits could be better taught using interactive numerical and symbolic simulation.

Pedagogic Examples of Simple Linear Circuits Problems

In the examples to follow, attention centers on the symbolic simulation results produced by Analog Insydes. We shall assume the reader knows how to use PSpice to obtain simple dc bias point .OP and linear frequency sweep (phasors) .AC analysis. Numerical (PSpice) and symbolic (Analog Insydes) simulation results are presented and some comparison are offered. The purpose of this paper is not to lay out specifics or technical details.

Example 1: Simple DC parallel resistor circuit

In the first example, Fig. 1, we have an ideal DC current source in parallel with three resistors. We will solve this example by using both PSpice and Analog Insydes and compare the results. An expression for V_1 is obtained; some results are given illustrating approximation working in conjunction with one of the resistors, R_3 , passing to a ∞ limit.

Fig. 1 Simple DC parallel resistor circuit.

Figure 2 depicts the output produced from PSpice .OP analysis, from this output file there is no relevant information produced to give us any insight into the functional dependencies between the circuit elements and the output behavior. The only information produced by the output file is that the voltage at node 1 is 4.8V.

From PSpice netlist file *.cir describing the circuit of Fig.1 we can use the built in function of Analog Insydes ReadNetlist to extract the netlist. We can also extract the schematic figure *.DXF with DXFGraphics. Fig. 3 shows what the results from Analog Insydes look like. From the figure we can show the advantages of using such an analytical analysis tool.

Symbolic analysis and computer algebra methods constitute a natural approach towards capturing the expertise of experienced designers. Symbolic analysis is a formal technique for calculating the behavior of circuits as closed form mathematical expressions. From Fig. 3 the formulas provide insight into the functional dependencies between the circuit elements and the output behavior, for example the voltage at node one is not just a number!

```

** Creating circuit file "ex1-schematic1-ex1.sim.cir"
*Libraries:
* Local Libraries :
* From [PSPICE NETLIST] section of pspice91.ini file:
.lib "nom.lib"
*Analysis directives: I
.OP
.PROBE
.INC "ex1-SCHEMATIC1.net"
**** INCLUDING ex1-SCHEMATIC1.net ****
* source EX1
I_Is      0 1 DC 2
R_R1      1 0 6
R_R2      1 0 8
R_R3      1 0 8
**** RESUMING ex1-schematic1-ex1.sim.cir ****
.INC "ex1-SCHEMATIC1.als"
**** INCLUDING ex1-SCHEMATIC1.als ****
.ALIASES
I_Is      Is(+0 -=1 )
R_R1      R1(1=1 2=0 )
R_R2      R2(1=1 2=0 )
R_R3      R3(1=1 2=0 )
-_(1=1)
.ENDALIASES
**** RESUMING ex1-schematic1-ex1.sim.cir ****
.END
**** SMALL SIGNAL BIAS SOLUTION      TEMPERATURE = 27.000 DEG C
*****
NODE   VOLTAGE   NODE   VOLTAGE   NODE   VOLTAGE   NODE   VOLTAGE
( 1)   4.8000
VOLTAGE SOURCE CURRENTS
NAME      CURRENT
TOTAL POWER DISSIPATION 0.00E+00 WATTS
**** OPERATING POINT INFORMATION      TEMPERATURE = 27.000 DEG C
*****
JOB CONCLUDED
TOTAL JOB TIME          .03

```

Fig. 2 PSpice output file for a .OP analysis of the circuit of Fig. 1.

```
<< AnalogInsydes`

SetDirectory["C:\\UTIL\\ASEE\\AI\\ex1\\DCparallelCKT-PSpiceFiles\\SCHEMATIC1\\DC"];

ParallelDC = ReadNetlist["DC.cir", KeepPrefix -> False, Simulator -> "PSpice"];
DisplayForm@netlist

Circuit:
Netlist (Raw, 4 Entries):
{I1, {0, 1}, Type -> CurrentSource, Value -> {AC -> 0, DC | Transient -> 2.}, Symbolic -> {DC | Transient -> I1}}
{R1, {1, 0}, Type -> Resistor, Value -> 6., Symbolic -> R1}
{R2, {1, 0}, Type -> Resistor, Value -> 8., Symbolic -> R2}
{R3, {1, 0}, Type -> Resistor, Value -> 8., Symbolic -> R3}
GlobalParameters[Simulator -> PSpice]

mma = CircuitEquations[ParallelDC, AnalysisMode -> DC, ElementValues -> Symbolic];
DisplayForm@mma


$$\left\{ \left\{ \frac{V\phi 1}{R1} + \frac{V\phi 1}{R2} + \frac{V\phi 1}{R3} = IS \right\}, \{V\phi 1\}, \text{DesignPoint} \rightarrow \{IS \rightarrow 2, R1 \rightarrow 6, R2 \rightarrow 8, R3 \rightarrow 8\} \right\}$$


sol = Solve[mma]


$$\left\{ \left\{ V\phi 1 \rightarrow \frac{IS R1 R2 R3}{R1 R2 + R1 R3 + R2 R3} \right\} \right\}$$


V$1 /. First@Solve[ApplyDesignPoint[mma, KeepSymbolic -> R1]]


$$\frac{8 R1}{4 + R1}$$


Limit[V$1 /. First[sol], R3 -> ∞]


$$\frac{IS R1 R2}{R1 + R2}$$


eqs = CircuitEquations[ParallelDC, AnalysisMode -> DC, ElementValues -> Symbolic, Formulation -> SparseTableau];
DisplayForm@eqs


$$\{ \{ V\phi IS + V\phi R1 = 0, V\phi IS + V\phi R2 = 0, V\phi IS + V\phi R3 = 0, \\ -I\phi IS + I\phi R1 + I\phi R2 + I\phi R3 = 0, I\phi IS = IS, I\phi R1 R1 - V\phi R1 = 0, I\phi R2 R2 - V\phi R2 = 0, I\phi R3 R3 - V\phi R3 = 0 \}, \\ \{ V\phi IS, V\phi R1, V\phi R2, V\phi R3, I\phi IS, I\phi R1, I\phi R2, I\phi R3 \}, \text{DesignPoint} \rightarrow \{ IS \rightarrow 2, R1 \rightarrow 6, R2 \rightarrow 8, R3 \rightarrow 8 \} \}$$


solutions = Solve[eqs]


$$\left\{ \left\{ I\phi R3 \rightarrow \frac{IS R1 R2}{R1 R2 + R1 R3 + R2 R3}, V\phi R3 \rightarrow \frac{IS R1 R2 R3}{R1 R2 + R1 R3 + R2 R3}, I\phi IS \rightarrow IS, I\phi R1 \rightarrow \frac{IS R2 R3}{R1 R2 + R1 R3 + R2 R3}, \right. \right. \\ \left. \left. V\phi R1 \rightarrow \frac{IS R1 R2 R3}{R1 R2 + R1 R3 + R2 R3}, I\phi R2 \rightarrow \frac{IS R1 R3}{R1 R2 + R1 R3 + R2 R3}, V\phi IS \rightarrow -\frac{IS R1 R2 R3}{R1 R2 + R1 R3 + R2 R3}, V\phi R2 \rightarrow \frac{IS R1 R2 R3}{R1 R2 + R1 R3 + R2 R3} \right\} \right\}$$


Limit[solutions, R3 -> ∞]


$$\{ V\phi IS \rightarrow -\frac{IS R1 R2}{R1 + R2}, V\phi R1 \rightarrow \frac{IS R1 R2}{R1 + R2}, V\phi R2 \rightarrow \frac{IS R1 R2}{R1 + R2}, V\phi R3 \rightarrow \frac{IS R1 R2}{R1 + R2}, I\phi IS \rightarrow IS, I\phi R1 \rightarrow \frac{IS R2}{R1 + R2}, I\phi R2 \rightarrow \frac{IS R1}{R1 + R2}, I\phi R3 \rightarrow 0 \}$$


```

Fig. 3 Analog Insydes analysis results of the circuit of Fig. 1

Example 2: CCCS, dc emitter follower

In the second example, Fig. 4, we demonstrate the analysis of a simple current control current source electric circuit. It shows the advantages of using analytic computer algebra to solve the circuit. PSpice output file and Analog Insydes results will be compared.

Fig. 4 Current controlled current source example, dc emitter follower.

Figure 5 is PSpice output file results of analyzing Fig. 4, again from output file the node voltages and currents can be read as results of simulation.

```

** Creating circuit file "ex5-SCHEMATIC1-ex5.sim.cir"
** WARNING: THIS AUTOMATICALLY GENERATED FILE MAY BE OVERWRITTEN BY SUBSEQUENT SIMULATIONS
*Libraries:
* Local Libraries :
* From [PSPIICE NETLIST] section of pspice91.ini file:
.lib "nom.lib"
*Analysis directives:
.OP
.PROBE
.INC "ex5-SCHEMATIC1.net"
**** INCLUDING ex5-SCHEMATIC1.net ****
* source EX5
F_F1      0 3 VF_F1 150
VF_F1     2 3 0V
R_Rpi     1 2 2k
R_RE      3 0 1k
I_Ix      0 3 DC 6
V_Vs      1 0 12
**** RESUMING ex5-SCHEMATIC1-ex5.sim.cir ****
.INC "ex5-SCHEMATIC1.als"
**** INCLUDING ex5-SCHEMATIC1.als ****
.ALIASES
F_F1      F1(3=0 4=3 )
VF_F1     F1(1=2 2=3 )
R_Rpi     Rpi(1=1 2=2 )
R_RE      RE(1=3 2=0 )
I_Ix      Ix(+0 -3 )
V_Vs      Vs(+1 -0 )
- _ (3=3)
- _ (1=1)
- _ (2=2)
.ENDALIASES
**** RESUMING ex5-SCHEMATIC1-ex5.sim.cir ****
.END
****
** Profile: "SCHEMATIC1-ex5" [ E:\pspice\dc\5\ex5-SCHEMATIC1-ex5.sim ]
**** SMALL SIGNAL BIAS SOLUTION TEMPERATURE = 27.000 DEG C
*****
NODE  VOLTAGE  NODE  VOLTAGE  NODE  VOLTAGE  NODE  VOLTAGE
( 1) 12.0000 ( 2) 90.2750 ( 3) 90.2750
VOLTAGE SOURCE CURRENTS
NAME      CURRENT
VF_F1     -3.914E-02
V_Vs      3.914E-02
TOTAL POWER DISSIPATION -4.70E-01 WATTS
****
** Profile: "SCHEMATIC1-ex5" [ E:\pspice\dc\5\ex5-SCHEMATIC1-ex5.sim ]
**** OPERATING POINT INFORMATION TEMPERATURE = 27.000 DEG C
*****
**** CURRENT-CONTROLLED CURRENT SOURCES
NAME      F_F1
I-SOURCE  -5.871E+00
JOB CONCLUDED
TOTAL JOB TIME .06

```

Fig. 5 PSpice Current Controlled Current Source Example Output.

Unlike PSpice simulation and as seen in Fig. 6 which is the Analog Insydes output file, we can see the advantages of using symbolic analysis. Here, we can see what each node voltage and branch current depends on. The input and output resistances can easily be calculated as shown.

```

Circuit:
Netlist (Raw, 5 Entries):
{F1, {2 → 1, 3 → 2, 0 → 3, 3 → 4}, Subcircuit → SCHEMATIC1_F1, Selector → Subcircuit}
{RE, {3, 0}, Type → Resistor, Value → 1000., Symbolic → RE}
{IX, {0, 3}, Type → CurrentSource, Value → {AC → 0, DC | Transient → 6.}, Symbolic → {DC | Transient → IX}}
{VS, {1, 0}, Type → VoltageSource, Value → {AC → 0, DC | Transient → 12.}, Symbolic → {DC | Transient → VS}}
{RPI, {1, 2}, Type → Resistor, Value → 2000., Symbolic → RPI}
Subcircuit:
Name → SCHEMATIC1_F1
Selector → Subcircuit
Ports → {1, 2, 3, 4}
Parameters → {F1}
Translation → {}
Definition →
Netlist (Raw, 2 Entries):
{F1, {1, F1$1, 3, 4}, Type → CCCSource, Value → 150., Symbolic → F1}
{F1, {F1$1, 2}, Type → VoltageSource, Value → {AC → 0, DC | Transient → 0}}
GlobalParameters[Simulator → PSpice]

mma = CircuitEquations[netlist, AnalysisMode → DC, ElementValues → Symbolic];
DisplayForm@mma

$$\left\{ \left\{ I_1 VS + \frac{V_1 - V_2}{RPI} = 0, I_1 F1 + \frac{-V_1 + V_2}{RPI} = 0, -F1 F1 I_1 F1 - I_1 F1 + \frac{V_3}{RE} = IX, -I_1 F1 + I_1 F1 = 0, \right. \right.$$


$$\left. V_2 - V_1 F1 = 0, -V_3 + V_1 F1 = 0, V_1 = VS \right\}, \{V_1, V_2, V_3, V_1 F1, I_1 F1, I_1 VS\},$$


$$\text{DesignPoint} \rightarrow \{F1 \rightarrow 150., RE \rightarrow 1000., IX \rightarrow 6., VS \rightarrow 12., RPI \rightarrow 2000., \text{Simulator} \rightarrow \text{PSpice}\}$$


sol = Solve[mma]

$$\left\{ \left\{ I_1 VS \rightarrow -\frac{-IX RE + VS}{RE + F1 F1 RE + RPI}, V_3 \rightarrow -\frac{-IX RE RPI - RE VS - F1 F1 RE VS}{RE + F1 F1 RE + RPI}, V_1 F1 \rightarrow -\frac{-IX RE RPI - RE VS - F1 F1 RE VS}{RE + F1 F1 RE + RPI}, \right. \right.$$


$$\left. I_1 F1 \rightarrow -\frac{IX RE - VS}{RE + F1 F1 RE + RPI}, V_2 \rightarrow -\frac{-IX RE RPI - RE VS - F1 F1 RE VS}{RE + F1 F1 RE + RPI}, I_1 F1 \rightarrow -\frac{IX RE - VS}{RE + F1 F1 RE + RPI}, V_1 \rightarrow VS \right\}$$


dp = GetDesignPoint[mma]
{F1 → 150., RE → 1000., IX → 6., VS → 12., RPI → 2000., Simulator → PSpice}

sol /. dp // N
{{I1 VS → 0.0391373, V3 → 90.2745, V1 F1 → 90.2745, I1 F1 → -0.0391373, V2 → 90.2745, I1 F1 → -0.0391373, V1 → 12.}}
```

Db = -I\$VS /. First[sol]

$$\frac{-IX RE + VS}{RE + F1 F1 RE + RPI}$$

Rin = 1/Limit[Limit[Db, IX → 0], VS → 1]

$$RE + F1 F1 RE + RPI$$

Vo = V\$3 /. First[sol]

$$-\frac{IX RE RPI - RE VS - F1 F1 RE VS}{RE + F1 F1 RE + RPI}$$

Rout = 1/Limit[Limit[Vo, VS → 0], IX → 1]

$$\frac{RE + F1 F1 RE + RPI}{RE RPI}$$

Fig. 6 Analog Insydes current controlled current source emitter follower example.

Example 3: RC low-pass filter example

In the last example, Fig. 7, we will investigate the high frequency response of a simple RC low-pass filter with values as shown in Figure. The circuit is a first-order low-pass filter. Since we want the gain of this filter, it is convenient to make the input voltage 1 volt so the output voltage is numerically equivalent to the gain. Alternatively, we can easily divide the input voltage into the output voltage.

Fig. 7 RC low-pass filter example.

After describing the circuit depicted in Fig. 7 and specifying the frequency range in Capture for an AC analysis, PROBE provides plots with several options for Functions or Macros. Here we are interested in obtaining the gain magnitude or getting gain in decibels, we choose “Add” from the “Trace” menu in PROBE. Then select the “DB” function in the right-hand column and choose “V(2)” from the left-hand column. The next display for this example is to have PROBE plot the phase shift as a function of frequency. We simply specify “VP(2)” from the “Add Trace” dialog box. PROBE automatically shows the phase shift in degrees.

Figure 8 shows the results of PSpice for the magnitude and phase. Note that there is no relevant information produced in the output file for this type of analysis.

Fig. 8 PSpice magnitude and Phase plots of the Low-pass filter of Fig. 7.

Again notice from Fig. 9, the Analog Insydes output, that the output voltage across the capacitor at V(2) is not just a plot as produced by PSpice, instead we have an equation that states what V(2) depends on; in this case, it consists of the input voltage VIN divided by $1 + C1 \cdot R1 \cdot s$. The output level varies from 1V at $f = 1$ Hz to almost 0V at $f = 1$ MHz. When the frequency is low, the value of X is large, allowing most of the source voltage of 1 V to appear across node 2. As the frequency increases, X becomes smaller and V(2) diminishes. Also, recall that we are dealing with phasors and that the two voltages VR and VC are always 90° apart. So, when the two voltages are equal in magnitude, $V_C = 0.707 \angle -45^\circ$ V, $V_R = 0.707 \angle 45^\circ$ and $V(2) = 0.707$ V. In AC analysis, this formula is easily shown to be $f_H = 1/(2 \cdot \pi \cdot R \cdot C) = 1.591$ kHz which is the point of

inflection on the phase-angle plot. This can easily be verified from the Bode phase plot of either the PSpice or Analog Insydes simulation used.

```
<< AnalogInsydes`
SetDirectory["d:\\pspice\\ac\\ex2\\"];
lpf = ReadNetlist["lpf.cir", KeepPrefix -> False, Simulator -> "PSpice"];
DisplayForm@lpf
Circuit:
Netlist (Raw, 3 Entries):
{R1, {1, 2}, Type -> Resistor, Value -> 100000., Symbolic -> R1}
{VIN, {1, 0}, Type -> VoltageSource, Value -> {AC -> 1., DC | Transient -> 0}, Symbolic -> {AC -> VIN}
{C1, {0, 2}, Type -> Capacitor, Value -> 1. x 10^-9, Symbolic -> C1}
GlobalParameters[TSTEP -> 0., TSTOP -> 1. x 10^-6, Simulator -> PSpice]

mna = CircuitEquations[lpf, AnalysisMode -> AC, ElementValues -> Symbolic];
DisplayForm@mna
```

$$\begin{pmatrix} \frac{1}{R1} & -\frac{1}{R1} & 1 \\ -\frac{1}{R1} & \frac{1}{R1} + C1 s & 0 \\ 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} VS1 \\ VS2 \\ ISVIN \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ VIN \end{pmatrix}$$

```
solutions = Solve[mna]
```

$$\left\{ \left\{ ISVIN \rightarrow -\frac{C1 s VIN}{1 + C1 R1 s}, VS2 \rightarrow \frac{VIN}{1 + C1 R1 s}, VS1 \rightarrow VIN \right\} \right\}$$

```
dp = MakeDesignPoint[ExpandSubcircuits[lpf]]
```

```
{R1 -> 100000., C1 -> 1. x 10^-9, TSTEP -> 0., TSTOP -> 1. x 10^-6, Simulator -> PSpice}
```

```
Vout = VS2 /. First[solutions]
```

```
hs = Vout /. dp
```

$$\frac{VIN}{1 + 0.0001 s}$$

```
VIN = 1
```

```
1
```

```
BodePlot[hs /. s -> 2. π * I * f, {f, 1., 1.^6}, MagnitudeDisplay -> Linear,
PlotPoints -> 100, PlotStyle -> RGBColor[1, 0, 0]]
```

Fig. 9 Analog Insydes results of simulating the low-pass filter of Fig. 7.

Conclusions and Challenges

The use of symbolic results offers an insight into the cause and effect relationship that exists within the circuit or system. Further, symbolic analysis produces an algebraic formula that identifies dominant terms or poles and zeroes of an approximate transfer function.

Current problems are complex and demand computer simulation and synthesis. There is no going back to a simpler time. There is much to be gained by the introduction of computer methods and much can be lost by postponing or introducing them in a limited, feeble, adjunctive way. We believe that if efficiently taught an inexperienced student can understand the analysis of a 100-node circuit just as eagerly as they can understand the analysis of a one-node circuit. In education, there can be new ways and we propose one such way here.

We claim that if both computer tools, numerical and symbolic circuit simulation are introduced and used right from the beginning of the first course of the circuit sequence, a synergistic relationship between basic understanding and analytical ability would naturally evolve. The program can overcome the main barrier to greater in-depth coverage—the difficult task of making the necessary calculations.

To make sure that such a plan is successful. Obviously, the instructor must first become a seasoned and experienced user of both programs before he will be able to guide the students use thereof. In our experience, such programs are reasonably robust but by no means foolproof; the teacher must be willing to seek help when needed, and there will probably be times when he will have to roll with the punches. The computer does not dictate how it will be used nor even suggest how it might best be used; only by intelligent effort and experimentation will the teacher discover that. How are examinations to be designed and administered? That's a good question, likely as take-homes, individual or group. There is room for creativity. Perhaps the biggest challenges are overcoming the fear of the unknown, forsaking a familiar but time worn mindset and breaking the grip of habit and tradition.

We advocate basing instruction in circuit analysis on computer simulation for exactly the same reason that a modern carpenter prefers to use electric saws and pneumatic nail drivers rather than conventional hammers and hand saws.

REFERENCES

- [1] Analog Insydes 2.1, Intelligent Symbolic Design Systems, © 2003 by ITWM, Online at: <http://www.analog-insydes.de/>
- [2] Mathematica 5.2, © 2005, Wolfram Research, Inc.
- [3] L. W. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits." Tech. Rep. UCB/ERL M520, University of California, Berkeley, 1975
- [4] J.-C. Thomassian, Interactive Symbolic and Numerical Analysis and Design of Circuits, Ph.D. dissertation, The University of Toledo, Ohio © 2002
- [5] J.-C. Thomassian and E. D. Smith, "Interactive Numerical and Symbolic Simulation: A New Paradigm for Teaching Circuits," 34th ASEE/IEEE *Frontiers in Education*, Session S2C pp. 8–12, October 20–23, 2004.
- [6] OrCAD PSpice 15.7, ® 2006, Cadence Design Systems, Inc.