International Journal of
Science and Engineering Investigations
vol. 2, issue 18, July 2013
ISSN: 2251-8843

A Functional Analysis of SPICE Simulations and Parameters

Karna D. Gajab[1], Ajay Kanth Chitturi[2]
[1]VLSI Professional, B.E. (Electronics and Communication, Gujarat Technological University)
[2]VLSI Professional, B.E. (Electronics and Instrumentation, BITS Pilani, Goa Campus)
([1]karna0490@gmail.com, [2]ajaykanthch@gmail.com)

Abstract- SPICE (*Simulation Program with Integrated Circuit Emphasis)* has always been a frontier for evaluation of designed circuits and predicting their behavior. From its original development at the Electronics Research Lab of the University of California, Berkeley (1975), it has upgraded and evolved into a widely used industry standard, competent to be termed as a powerhouse for general purpose analog circuit simulation. As the IC industry these days is heavily dependent on SPICE models of their designs and due to multiple vendors pitching in the market, usage of SPICE has grown exponentially. For designers and modeling engineers to effectively design and validate the circuit performance, it is vital to have a functional understanding of how SPICE simulations are carried out in the background and what are its essential dependencies. This paper attempts to highlight the fundamental working of SPICE algorithms, analysis types and simulation parameters so as to provide a comprehensive analysis of SPICE simulations and its internal operational mechanism as compared to conventional SPICE tutorials and references.

Keywords- *SPICE; simulation; analysis parameters*

I. Introduction

The fundamental element that enables simulations in SPICE is the SPICE 'engine' or typically the SPICE solver. It is responsible for finding solutions to the formulated circuit equations. In the earlier SPICE version i.e. SPICE2, it was mostly batch mode (*command line based*). Also it was case sensitive and the processing of simulation results was tedious. In SPICE2, results from the SPICE engine were scheduled to an output file which was then utilized as an input for the engine. With realizable advancements, SPICE3 was proposed. It is interactive, capable of multi-processing simulations and not case sensitive. Industry today mainly relies on the commercial GUI (*Graphical User Interface*) releases of SPICE e.g. *PSPICE* (based on *SPICE-2G7*), *LTSPICE*, *NGSPICE* and many other releases. The key reason behind studying the internal SPICE mechanism is understanding the simulator behavior in scenarios where the simulation might lead to erroneous results or fail to converge i.e. fail to calculate node voltages or branch currents. Another factor that plays a major role is that of SPICE simulation parameters or analysis parameters which serve as an all level check on the voltages and currents and verify that they are within acceptable limits or yielding suitable results for the specific circuit. Further sections of this paper provide an insight to the SPICE algorithm and illustrate the vitality of the simulation parameters.

II. BIAS POINT CALCULATION & SIMULATION PROCESS

A. *Solving circuit equations using the matrix approach*

The basic SPICE operation deals with formation of nodal equations for the circuit under review. The fundamental network law of KCL helps to form the nodal equations for the circuit. Let us assume that for the network shown below, the value of the current source and resistances are as follows: *R1 = 5Ω, R2 = 10Ω, R3 = 5 Ω, R4 = 10 Ω and I1 = 5A.*

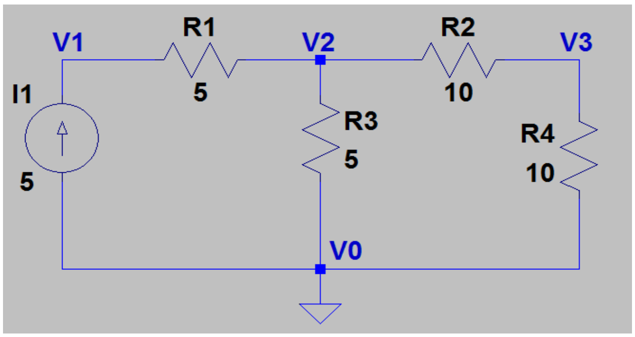


Fig 1: A simple R-Network with an independent current source

The nodal equations formed based on KCL will be:

$$\frac{V1-V2}{5}-5=0\,;\quad \frac{V2-V1}{5}+\frac{V2}{5}+\frac{V2-V3}{10}=0;\quad \frac{V3-V2}{10}+\frac{V3}{10}=0$$

The above KCL equations are then rearranged in the matrix form as follows:

$$\begin{bmatrix} 0.2 & -0.2 & 0 \\ -0.2 & 0.5 & -0.2 \\ 0 & -0.1 & 0.2 \end{bmatrix}\begin{bmatrix} V1 \\ V2 \\ V3 \end{bmatrix}=\begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix} \qquad \text{- (1)}$$

The above matrix equation is then reduced further by Gaussian Elimination. It can be inferred that the current matrix generally contains a zero entry except in cases wherein an active device like a current source is present. The current matrix is always a known entitiy and thus SPICE always solves nodal equations using the conductance matrix using the relation *[G] × [V] = [I]* , where [G] is the conductance matrix. The voltage matrix might not be known always and thus, [R]×[I]=[V] is not utilised. In case of voltage sources in the circuit, the Norton equivalent is calculated and then the matrix calculation is carried out. In case of DC bias point calculations, capacitors are open circuited and the inductors are short circuited.

B. The SPICE solver algorithm and types of analysis

For any simulation to be carried out, the SPICE engine has a supportive algorithm to ensure the functionality. The figure below illustrates the most fundamental SPICE algorithm that operates in the backend of every SPICE simulation:

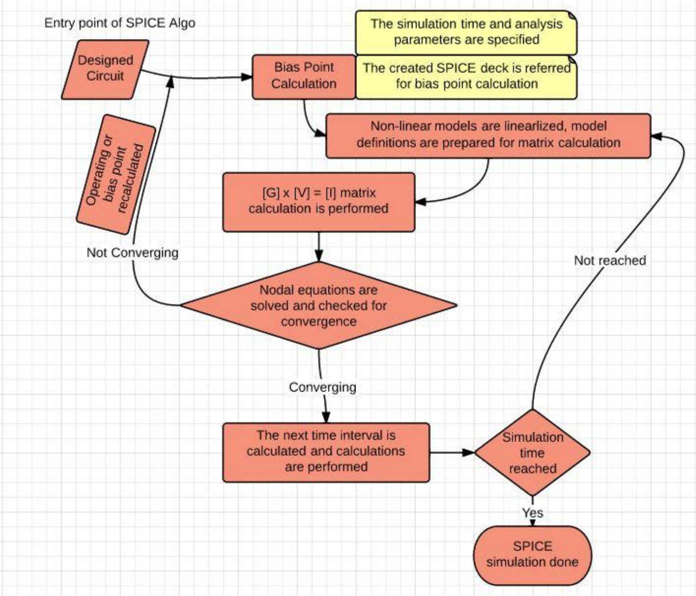

Entry point of SPICE Algo
Designed Circuit
Bias Point Calculation
The simulation time and analysis parameters are specified
The created SPICE deck is referred for bias point calculation
Non-linear models are linearlized, model definitions are prepared for matrix calculation
[G] x [V] = [I] matrix calculation is performed
Operating or bias point recalculated
Not Converging
Nodal equations are solved and checked for convergence
Not reached
Converging
The next time interval is calculated and calculations are performed
Simulation time reached
Yes
SPICE simulation done

Fig 2: A simplified flowchart of the SPICE Algorithm

As the above flowchart clearly illustrates all the unseen steps happening in the simulation flow, SPICE has the capability to perform different types of analysis on the circuits under test viz. a) DC Analysis b) Time response-Transient analysis c) Frequency response-AC analysis.

DC analysis is primarily focused on calculation of the DC bias or operating point as discussed earlier. A bias point is essential for time response and frequency response analysis. As stated earlier, capacitors and inductors and other energy storage elements are ignored during DC simulation. DC sweep aims at performing bias point calculation while sweeping a parameter of any component in a defined range within acceptable limits. For nonlinear circuits, linearized models effective to be termed as tantamount to the nonlinear models are created and used during simulation. When the fluctuations in the node voltages or branch currents are within acceptable limits, the results are said to be converged. The mathematical concept behind the same is that of numerical methods like Newton-Raphson method, discussed in more detail in the next section.

AC analysis performs the DC bias point calculation as the preliminary step, ignoring the energy storage elements. The nodal matrix with the complex impedances is loaded and non-linear models are converted to their equivalent small signal models. The process is then repeated for all frequencies specified for simulation and results are captured as magnitude and phase entities for each node voltage.

For large signal transient simulations, the entire nonlinear models are used and are often complex simulations. The open circuit and short conditions cannot be achieved for energy storage elements like capacitors and inductors and thus, numerical integration is heavily used with the nonlinear circuit.

III. NONLINEAR DC ANALYSIS AND NEWTON-RAPHSON ALGORITHM

A. Nonlinear DC Analysis

If we consider nonlinear circuits e.g. circuits containing nonlinear devices like diodes, Gaussian elimination cannot be performed for matrix calculations, the reason being transcendental nodal equations, which cannot be solved by numerical methods. For the circuit shown below, we will transform the nonlinear diode component to be utilized in nodal equation calculations.

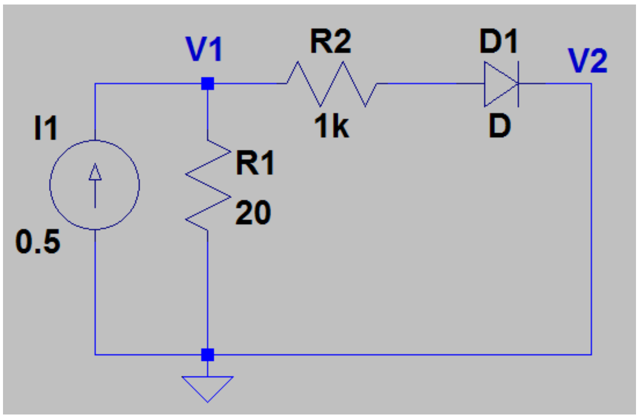


Fig 3: Circuit with D1 diode as a nonlinear component

It's known that the current-voltage relationship for a diode is defined as: $Id = Is(e^{\left(\frac{Vd}{Vt}\right)} - 1)$ where Is defines the saturation current of the diode and Vt denotes the thermal voltage. For analysis, consider the following diode characteristics curve:

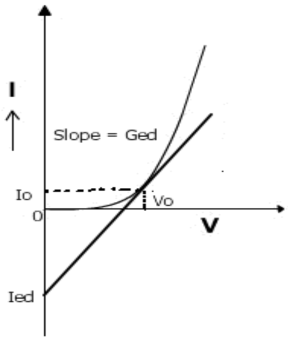


Fig 4: Diode forward characteristics curve with slope indicated

The slope of the characteristics can be denoted by the relation: $Ged = \frac{d(Id)}{d(Vd)} = \left(\frac{Is}{Vt}\right) e^{\left(\frac{Vo}{Vt}\right)}$ – (2). Also the tangent to the characteristics intercepts the y-axis at Ied and leads to the following relation: $Ied \;=\; Io - Ged.Vo$ – (3) which permits us to transform the nonlinear diode model into its linear equivalent as shown in the below figure:

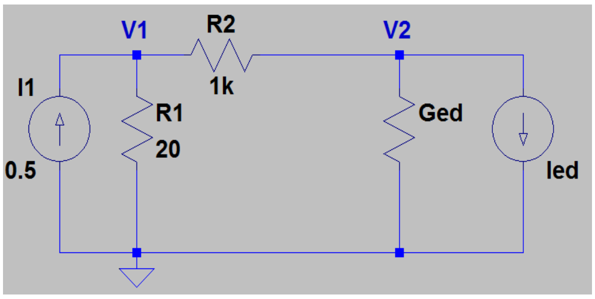


Fig 5: Linear model equivalent replaced for the nonlinear diode

As the circuit now contains all linear models, nodal equations are calculated as:

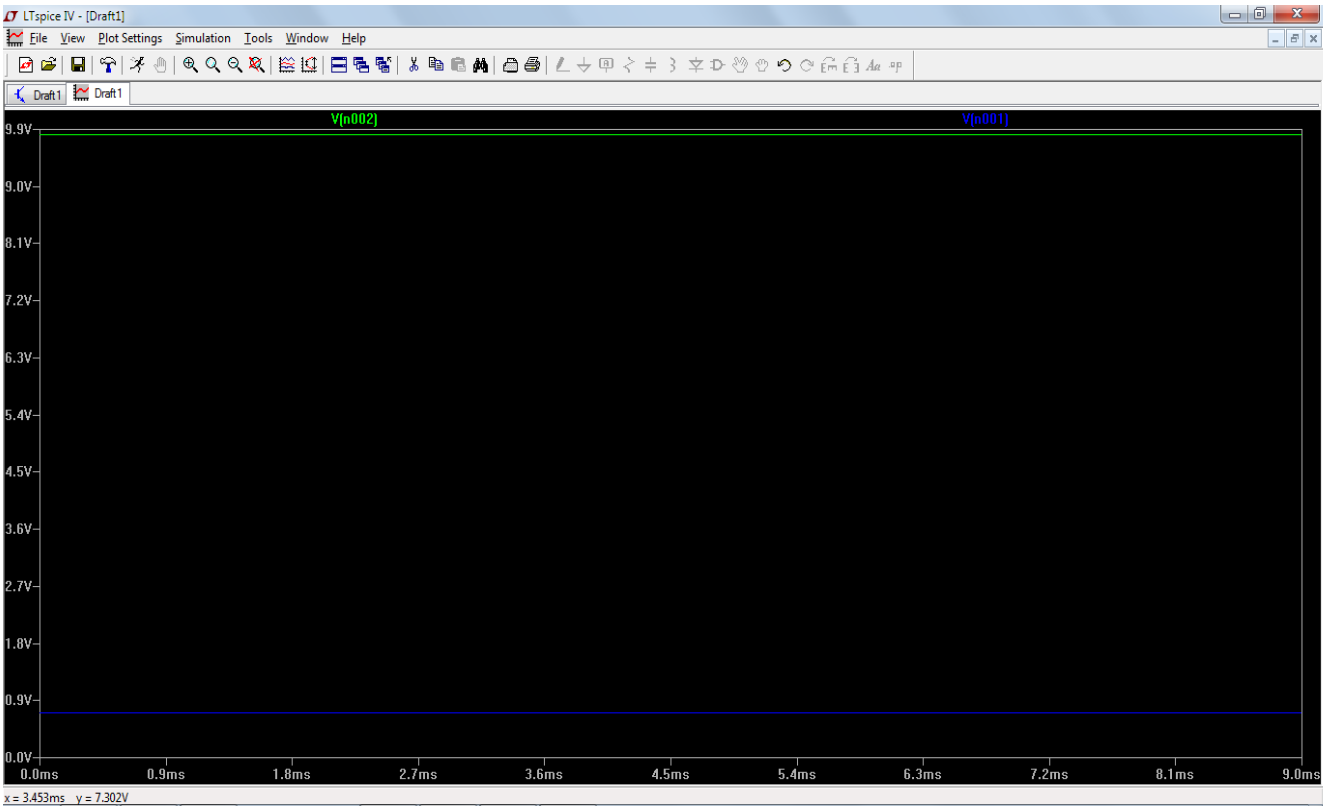
$$\left(\frac{1}{R1} + \frac{1}{R2}\right).V1 + V2.\left(-\frac{1}{R2}\right) = I1 = \left(\frac{1}{20} + \frac{1}{1000}\right).V1 + V2.\left(-\frac{1}{1000}\right) = 0.5 \qquad (4)$$

$$V1.\left(-\frac{1}{R2}\right) + \left(\frac{1}{R2} + Ged\right).V2 = \; -(Ied) = V1.\left(-\frac{1}{1000}\right) + (1/1000 + Ged).V2$$

The nodal analysis is performed only after the diode nonlinear model is converted to its linear equivalent at a particular bias point. As the initial point is unknown, it is generally a guess value used by the simulator. Once the nodal equations are solved, the next value of bias point is obtained and is used to calculate the linear equivalent model once more. This iteration is repeated until it has converged to a value. Deciding on whether the solution or value is reached or not, the changes in consecutive iteration values of voltages and currents are matched with limiting parameters. This method is nothing but the Newton-Raphson algorithm for computing iterations for calculation of the operating point.

B. Newton-Raphson iteration algorithm

The Newton-Raphson algorithm is a very powerful tool to compute solutions of equations numerically. It is fundamentally based on the concept of linear approximation. Let us consider $V(x)$ to be a well-defined voltage function and r be the root of the voltage function. An initial estimate of x_0 is chosen and then used to produce a better estimate x_1, which is then further chosen to compute x_2 as the next estimate and so on till the result is very close to r and cannot be iterated further. As a good estimate, assume: $r = \; x_0 + h$, where h is the measure of separation between the real root and the estimate. As the value of h is small, by tangent approximation we can say that: $h \;\approx\; -\frac{V(x_0)}{V'(x_0)}$ and $r = x_0 - \frac{V(x_0)}{V'(x_0)} = x_1$, the next estimate. Continuing this process results in a generalized expression for the algorithm stated below:



x = 3.453ms y = 7.302V

Fig.6 Simulation results for the circuit shown in Fig.5 in LTSpice

$$V_{n+1} = V_n - \frac{V(x_n)}{V'(x_n)} \qquad -(6)$$

The same algorithm can be realized graphically by taking $y = V(x)$ as the voltage curve meeting the x axis at point r.

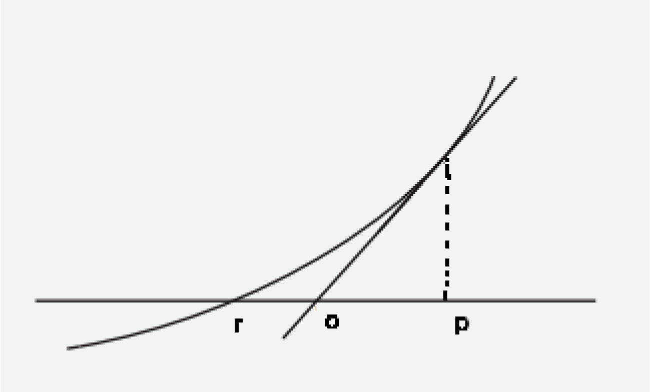


Fig 7: Graphical interpretation of the Newton-Raphson algorithm

The highlighted point o is the next estimate for r which is derived from the tangent line from the first estimate p, crossing the x axis. This procedure is iterated till the closest approximation to r is reached.

IV. CONVERGENCE PROBLEMS IN SPICE AND ANALYSIS PARAMETERS

It so happens that modeling and simulation engineers often face situations with their models wherein the circuit fails to converge to an output value. This especially happens when the circuits designed are having a lot of dependencies, more components and also in models which are hierarchical.

A. Factors determining convergence problems

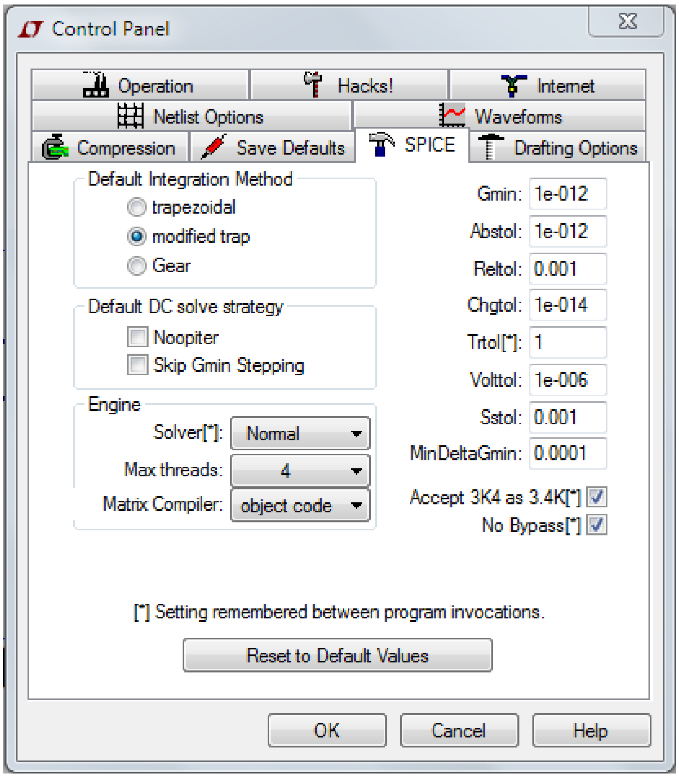
The understanding behind the convergence problem in SPICE is that Newton-Raphson repeated iterations fail to land up on consistent set of voltages and currents. The algorithm guarantees convergence provided certain conditions are met viz. a) Nonlinear equations have a solution b) Equations are continuous and derivatives for the equations can be computed c) Initial approximation is not very far away from the solution. So as to solve the convergence issues in spice, analysis parameters or simulation parameters play a very vital role.

B. Analysis/Simulation parameters in SPICE

There certainly exists a tradeoff between the simulation time for the circuit and the accuracy of the voltage and current results obtained. In order to ensure that the tradeoff is minimized and the results are as per the expectation of the modeling engineer, SPICE provides certain limiting parameters for the user to set for the particular simulation to be carried out. These parameters are termed as *analysis parameters or SPICE simulation parameters*. These parameters have significantly evolved over the version upgrades of SPICE over the years. The table below lists some prominent SPICE analysis parameters that are generally utilized to minimize the tradeoff and resolve convergence issues:

TABLE I. SPICE Analysis Parameters

Sr.No.	***Notable Spice Simulation/Analysis Parameters and their utility***		
	Parameter	***Funtionality***	***Default value in SPICE simulators***
1	*VNTOL*	*Defines the absolute voltage error tolerance for the simulation*	*1 µV*
2	*ABSTOL*	*Defines the absolute current error tolerance of the simulation*	*1 pA*
3	*RELTOL*	*Defines the relative error tolerance of the circuit*	*0.001 = 0.1%*
4	*PIVTOL*	*Defines the absolute minimum value for a matrix entry to be accepted as a pivot.*	*1.0e-13*
5	*TRTOL*	*Defines the transient error tolerance*	*7.0*
6	*CHGTOL*	*Defines the charge tolerance for the simulation*	*1.0e-14*
7	*ITL1*	*Denotes the DC iteration limit*	*100*
8	*ITL2*	*Denotes the DC transfer curve iteration limit*	*50*
9	*ITL4*	*Defines the transient analysis time point iteration limit*	*10*
10	*TEMP*	*Operating temperature of the circuit under simulation*	*300K (27^o C)*



Default Integration Method
○ trapezoidal
● modified trap
○ Gear

Default DC solve strategy
□ Noopiter
□ Skip Gmin Stepping

Engine
Solver[*]: Normal
Max threads: 4
Matrix Compiler: object code

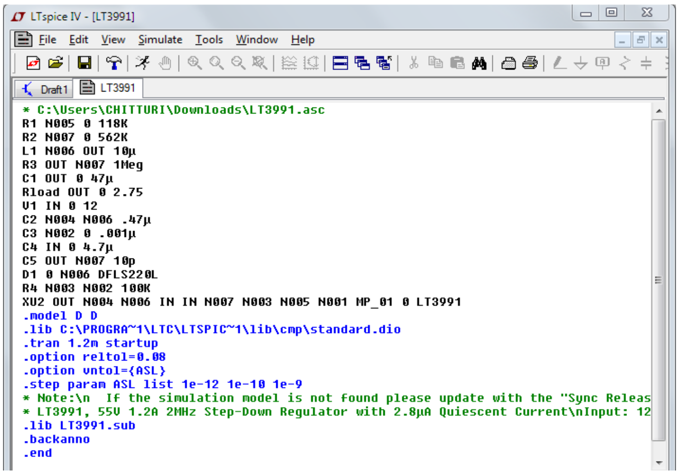
Gmin: 1e-012
Abstol: 1e-012
Reltol: 0.001
Chgtol: 1e-014
Trtol[*]: 1
Volttol: 1e-006
Sstol: 0.001
MinDeltaGmin: 0.0001
Accept 3K4 as 3.4K[*] ☑
No Bypass[*] ☑

[*] Setting remembered between program invocations.

Fig 8: Sim Control panel in LTSPICE with analysis parameters

V. EFFECT OF ANALYSIS PARAMETERS ON SIMULATION RESULTS IN LTSPICE

In order to illustrate the effects of analysis parameters, we make use of a *Linear Technology* model: *LT3991*, 55V, 1.2A Step-Down Regulator with 2.8 μA Quiescent current. The SPICE model for the component is available on Linear Technology's website for download. The netlist file with a *.cir* extension is used to simulate the model behavior.

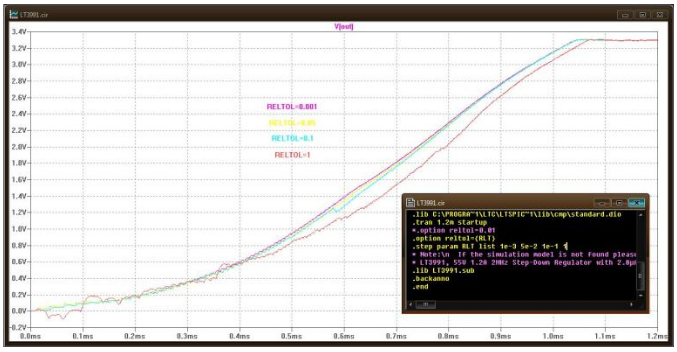


```
* C:\Users\CHITTURI\Downloads\LT3991.asc
R1 N005 0 118K
R2 N007 0 562K
L1 N006 OUT 10µ
R3 OUT N007 1Meg
C1 OUT 0 47µ
Rload OUT 0 2.75
V1 IN 0 12
C2 N004 N006 .47µ
C3 N002 0 .001µ
C4 IN 0 4.7µ
C5 OUT N007 10p
D1 0 N006 DFLS220L
R4 N003 N002 100K
XU2 OUT N004 N006 IN IN N007 N003 N005 N001 MP_01 0 LT3991
.model D D
.lib C:\PROGRA~1\LTC\LTSPIC~1\lib\cmp\standard.dio
.tran 1.2m startup
.option reltol=0.08
.option vntol={ASL}
.step param ASL list 1e-12 1e-10 1e-9
* Note:\n  If the simulation model is not found please update with the "Sync Releas
* LT3991, 55V 1.2A 2MHz Step-Down Regulator with 2.8µA Quiescent Current\nInput: 12
.lib LT3991.sub
.backanno
.end
```

Fig 9: Netlist file for the LT3991 model in LTSPICE

The output voltage for the model is 3.3V and in order to see the variation in the output voltage waveform with basic analysis parameters, we take 3 cases of variation in *RELTOL*, *VNTOL* and *ABSTOL*. The output voltage waveform for the regulator is then assessed for all the three stated cases.*RELTOL* is the relative voltage and current tolerance for the node voltages and the branch currents respectively. If the voltage at every node of the simulation is changed in value by less than *RELTOL* percentage of the iteration values, we can state that the node voltages are converging to a solution. For very small values of the node voltages, *VNTOL* plays a key role, which is nothing but the absolute node voltage tolerance. Once the simulation results for the node voltages are within limits of one of the above parameters, our final solution is said to have converged. *ABSTOL*, which is absolute branch current tolerance, plays the same role for branch current iteration values as *VNTOL*.

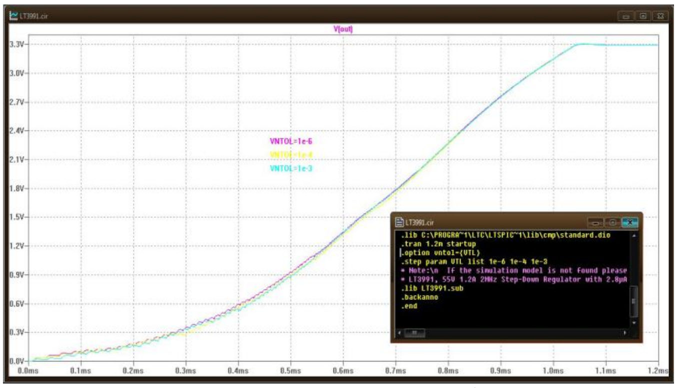
A. *Variation in output voltage wabveform for LT3991 with change in RELTOL values*



```
.lib C:\PROGRA~1\LTC\LTSPIC~1\lib\cmp\standard.dio
.tran 1.2m startup
*.option reltol=0.01
.option reltol={RLT}
.step param RLT list 1e-3 5e-2 1e-1 1
* Note:\n  If the simulation model is not found please
* LT3991, 55V 1.2A 2MHz Step-Down Regulator with 2.8µA
.lib LT3991.sub
.backanno
.end
```

Fig 10: Reltol variation in the output voltage waveform (0.001 to 1)

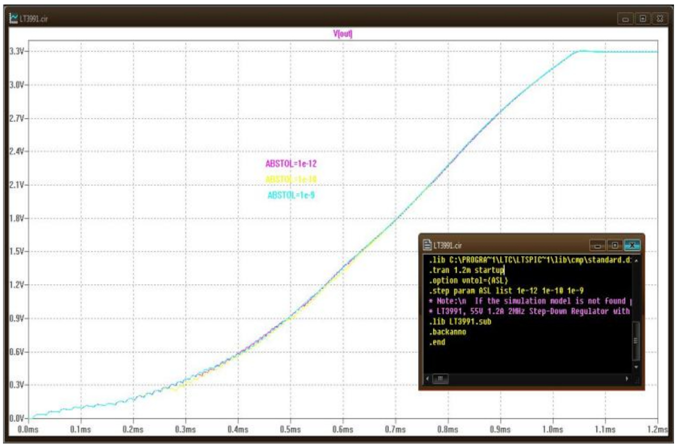
B. *Variation in output voltage wabveform for LT3991 with change in VNTOL values*



```
.lib C:\PROGRA~1\LTC\LTSPIC~1\lib\cmp\standard.dio
.tran 1.2m startup
.option vntol={VTL}
.step param VTL list 1e-6 1e-4 1e-3
* Note:\n  If the simulation model is not found please
* LT3991, 55V 1.2A 2MHz Step-Down Regulator with 2.8µA
.lib LT3991.sub
.backanno
.end
```

Fig 11: Vntol variation in the output voltage waveform (1e-6 to 1e-3)

C. *Variation in output voltage wabveform for LT3991 with change in ABSTOL values*



```
.lib C:\PROGRA~1\LTC\LTSPIC~1\lib\cmp\standard.d
.tran 1.2m startup
.option vntol={ASL}
.step param ASL list 1e-12 1e-10 1e-9
* Note:\n  If the simulation model is not found
* LT3991, 55V 1.2A 2MHz Step-Down Regulator with
.lib LT3991.sub
.backanno
.end
```

Fig 12: Abstol variation in the output voltage waveform (1e-12 to 1e-9)

Once the steady-state voltage of 3.3V is reached, the regulator voltage waveform does not show much deviation in the result. Even during the transient period of voltage rise, the *VNTOL and ABSTOL variation* curves do not show much distinction in the calculated voltage values and the solution is converging. But as *RELTOL* tolerance limit is also as indicator for the converging values of the node voltages and branch currents, the *RELTOL* variation curve shows deviation in the results as the value of *RELTOL* is increased from *1m* to *1*. The transient period of the voltage rise shows distinct deviation of the output voltage at higher values of *RELTOL*. It is therefore seen that many of these basic analysis parameters have a combined effect on the simulation results and adjusting the right parameters can help in eliminating convergence issues and yielding accurate results for the circuit model.

VI. LIMITATIONS OF SPICE

Despite its numerous positivity's, for complex circuits, the simulation results can be a bit misleading as *SPICE* is free from interference, noise and does not take these effects into account. Also in a complex circuit environment, crosstalk is a natural phenomenon which *SPICE* simply ignores. The real silicon behavior of the circuit has numerous other fabrication led issues which cannot be imported into the simulation environment, but only predicted within the tolerance limit. Also the tradeoff discussed earlier in this paper can prove a hindrance in the model results. Choosing accuracy over simulation time might not always be advisable. Analyzing the results from *SPICE* simulations in a suitable manner, keeping in mind the real world effects, can definitely prove *SPICE* simulations to be great use for model designers and validation engineers to comprehend and predict their model outcomes.

VII. CONCLUSION

In this paper, the *SPICE* simulation algorithm and much of the unknown simulation mechanism was discussed. The matrix calculation in the backend and numerical methods for calculation of node voltages and branch currents were discussed. Different analysis types and nonlinear DC analysis in particular was discussed with a suitable example circuit. Convergence issues in *SPICE,* the Newton-Raphson algorithm being the backbone for bias point calculation and how analysis parameters help in eliminating convergence were dealt in the later part of the paper. Finally, effects of variation in *RELTOL, VNTOL and ABSTOL* were observed for *LT3991, a step down regulator model of Linear Technology.* Thus, this paper gives a comprehensive understanding of *SPICE* simulation mechanism and *SPICE* parameters that control the simulation.

REFERENCES

[1] Ron Keilkowski, *Inside SPICE* McGraw Hill, 1994, ISBN 0-07-913712-1J.

[2] Andrei Vladimirescu, *The Spice Book, John Wiley & Sons, Inc., 1994, ISBN 0-471-60926-9.*

[3] Lawrence W. Nagel, *SPICE2: A Program to Simulate Semiconductor Circuits,* Memo UCB/ERL M520, 1975, University of California, Berkeley, CA.

[4] W. Banzhaf, Computer-Aided Circuit Analysis Using SPICE, Prentice Hall, Eaglewood Cliffs, NJ, 1989.

[5] P.W. Tuinenga, *SPICE, A guide to Circuit Simulation and Analysis Using PSpice,* Prentice-Hall, Eaglewood Cliffs, NJ, 1988.

[6] Linear Technology Corporation, *LT3991/LT3991-3.3/LT3991-5, 55V, 1.2A Step-Down Regulator with 2.8uA Quiescent current,* 1630 McCarthy Blvd., Milpitas, CA 95035-7417

[7] Zhou, T.Y; Texas Instrum., Dallas, TX, Zhou , Hua Zhang ,Xinyue Niu, *Foundational-circuit-based spice simulation, IEEE,* Circuits and Systems, 2008. ISCAS 2008, 978-1-4244-1684-4.

Karna D. Gajab was born in India, in 1990. He obtained his B.E. in Electronics and Communication engineering from Gujarat Technological University in 2012. His research interests include Computational & Simulation methods and embedded systems. Mr. Karna is an active member of IEEE.

Ajay Kanth Chitturi was born in India, in 1988. He obtained his B.E. in Electronics and Instrumentation engineering from Birla Institute of Technology and Science, Pilani – Goa Campus. His research interests include modeling and computer simulations.