SAND REPORT

SAND2004-2283
Unlimited Release
Printed June, 2004

Xyce™ Parallel Electronic Simulator Design

Mathematical Formulation, Version 2.0

Eric R. Keiter, Scott A. Hutchinson, Robert J. Hoekstra, Thomas V. Russo, and Lon J. Waters

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/ordering.htm

SAND2004-2283
Unlimited Release
Printed June, 2004

Xyce™ Parallel Electronic Simulator Design

Mathematical Formulation, Version 2.0

Eric R. Keiter, Scott A. Hutchinson, and Robert J. Hoekstra
Computational Sciences

Thomas V. Russo and Lon J. Waters
Component Information and Models

Sandia National Laboratories
P.O. Box 5800
Mail Stop 0316
Albuquerque, NM 87185-0316

Abstract

This document is intended to contain a detailed description of the mathematical formulation of **Xyce**, a massively parallel SPICE-style circuit simulator developed at Sandia National Laboratories. The target audience of this document are people in the role of "service provider". An example of such a person would be a linear solver expert who is spending a small fraction of his time developing solver algorithms for **Xyce**. Such a person probably is not an expert in circuit simulation, and would benefit from an description of the equations solved by **Xyce**. In this document, modified nodal analysis (MNA) is described in detail, with a number of examples. Issues that are unique to circuit simulation, such as voltage limiting, are also described in detail.

Acknowledgements

The authors would like to acknowledge the entire Sandia National Laboratories HPEMS (High Performance Electrical Modeling and Simulation) team, including Carolyn Bogdan, Regina Schells, Ken Marx, Steve Brandon, David Shirley and Bill Ballard, for their support on this project. We also appreciate very much the work of Becky Arnold, Mike Williamson, Brett Bader and Phil Campbell for help in reviewing this document.

Trademarks

Contacts

Bug Reports	http://tvrusso.sandia.gov/bugzilla
Email	xyce-support@sandia.gov
World Wide Web	http://www.cs.sandia.gov/Xyce

Contents

This page is left intentionally blank

Figures

This page is left intentionally blank

1 Introduction

This document describes how circuit problems are formulated and solved in **Xyce**, a massively parallel analog circuit simulator. This document was motivated by the need to address common questions, asked by people new to the **Xyce** research/development effort. The prerequisite for understanding this document is some experience in numerical simulation, but not necessarily circuit simulation.

This is not intended to be an exhaustive treatment of circuit theory. What is presented here is a detailed summary (with examples) of how circuit problems are posed in **Xyce**. This includes how the problem is formulated, what equations are solved, and some of the techniques for obtaining the solution. There are many time integration, nonlinear and linear solver solution techniques available inside of **Xyce**, and most of them are not described in this document. The solution techniques described have been limited to those for which one or more of the following is true:

1. The technique changes the set of equations being solved.
2. The technique is (apparently) unique to circuit simulation.
3. The technique is not well described by the literature.

One such technique, voltage limiting, is described in the nonlinear solver section. Another such technique, state variable condensation, is described in the time integration section.

2 Basics

If viewed abstractly, transient circuit problems are solved similarly to transient implicit partial differential equation (PDE) problems. There are three nested solvers: a time integrator, a nonlinear solver, and a linear solver. The relationship between the three nested solvers is illustrated in Figure 1.

Like a PDE problem, a circuit problem is based upon a topology. However, unlike a PDE problem, the topology is derived from an arbitrary circuit network connectivity, rather than a mesh. As such, the circuit topologies are generally much more heterogeneous than mesh-based PDE topologies. Strang [13] describes some of the analogies between circuits and PDE problems.

3 Circuit Problems

In this section, Kirchhoff's Laws, are described. Also, Modified Nodal Analysis (MNA) is introduced. The end of this section includes a linear circuit example.

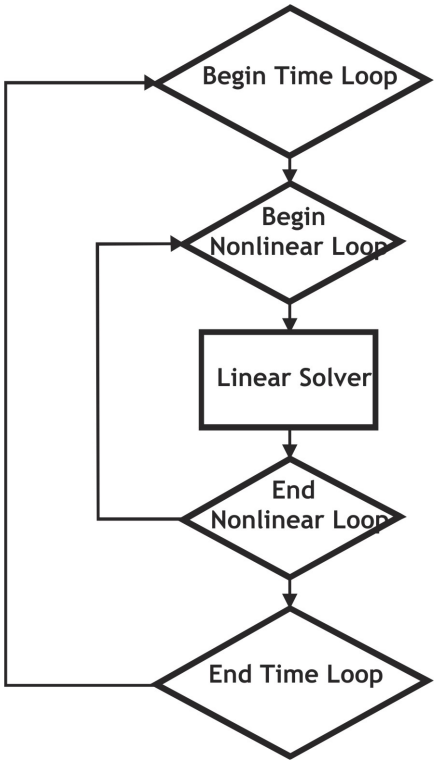

Figure 1. Xyce solver structure.

3.1 Kirchhoff's Laws

Circuit networks are subject to Kirchhoff's Current Law (KCL) and Kirchhoff's Voltage Law (KVL). Kirchhoff's current law specifies that at any node in a circuit the sum of the branch currents into/out of the node must equal zero. This is expressed by equation (1).

$$\sum_{i=0}^{N_1} I_i = 0, \tag{1}$$

where N_1 is the number of branch currents into/out of a circuit node. An illustration this equation for a single circuit node can be found in Figure 2. Equation (1) enforces the conservation of charge, as current is a measure of how much charge flows through a wire. It is also equivalent to stating that the divergence of current around a circuit node equals zero. Equation 1 will hold for every node in a circuit. This naturally leads to a set of coupled simultaneous equations, one for each circuit node.

Kirchhoff's voltage law (KVL) states that the sum of the branch voltage drops around a closed loop of a circuit should equal zero. A graphical representation of KVL can be found in Figure 3. In the figure, there are four circuit nodes, and three closed loops. The three

loops are defined by nodes (1,2,4), nodes (2,3,4), and nodes (1,2,3,4). As with KCL, KVL can lead to a large set of coupled simulataneous equations, similar to Equations 3- 5.

$$\sum_{i=0}^{B_1} V_i = 0 \tag{2}$$

$$V_{12} + V_{24} + V_{41} = 0 \tag{3}$$

$$V_{23} + V_{34} + V_{42} = 0 \tag{4}$$

$$V_{12} + V_{23} + V_{34} + V_{41} = 0 \tag{5}$$

where B_1 is the number of branches in a closed loop. The subscripts used in Equations 3- 5 denote the two nodes defining the respective branch.

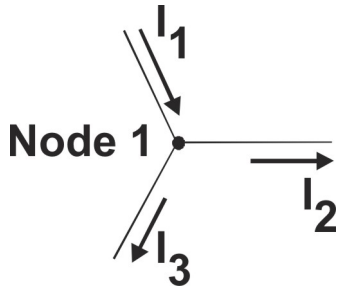

Figure 2. KCL for a single circuit node.

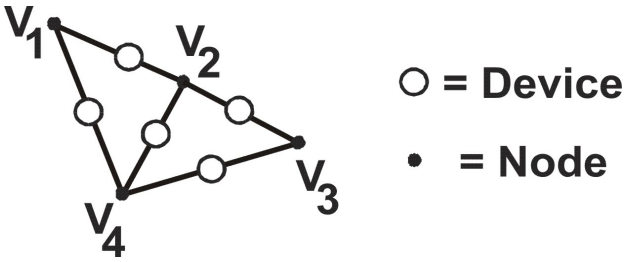

Figure 3. KVL for closed circuit loops. Equations 3- 5 correspond to this figure.

There are many different ways of formulating a system of equations to solve circuit problems. Some formulations, like the tableau formulation, explicitly include a full set of KCL and KVL equations. However, it is possible to combine KCL and KVL laws into a compact formulation because most branch currents are directly functions of their respective branch voltages. The formulation described in the next section, the modified KCL formulation, doesn't include any explicit KVL equations, as they are implicitly enforced by the formulation.

3.2 The modified KCL formulation

Circuit problems are usually solved on a computer using the “modified KCL formulation”. This is also sometimes referred to as modified nodal analysis (MNA). This is the formulation used in all of the common circuit simulators, such as Spice3f5, as well as **Xyce**. Modified nodal analysis, as well as several other types of circuit analysis, is described in detail by Vlach [14] and Chua [6]. The original paper describing this technique is by Ho [12] .

To best describe the modified KCL formulation, the *unmodified* KCL formulation (henceforth the KCL formulation) will be described first. In the KCL formulation, for every node (except the ground node) of the circuit, one KCL equation must be satisfied. For a circuit of N nodes, there will be a minimum of $N-1$ equations. There will also be a voltage variable for each node of the circuit, resulting in a minimum of $N-1$ variables. Most currents between circuit nodes can be expressed as a function of the voltage drop between the nodes. The simplest example of this is the current through a linear resistor. Consider the resistor in Figure 4, which is connected between 2 circuit nodes.

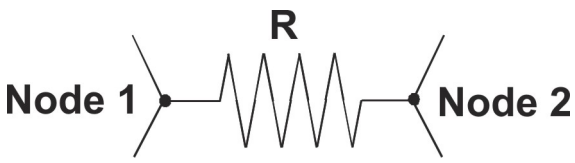

Figure 4. Resistor device.

The current through this resistor is defined by Ohm’s law, or $I = G \cdot V_{12}$. G is the conductance of the resistor (or $1/R$, where R is the resistance of the resistor), I is the current through the resistor, and $V_{12} = V_1 - V_2$ is the voltage drop across the resistor.

Most currents in a circuit can be expressed using this same type of expression ($I = G \cdot V$, or more generally $I = G(V) \cdot V$), in which currents are a function of voltage. Devices whose currents can be expressed in this way are also sometimes said to have *conductance* or, in the case of small signal analysis, *admittance* representations. Likewise, a matrix that consists of nothing but conductance (G) terms is sometimes referred to as a *conductance matrix* or (respectively) an *admittance matrix*. In the circuit simulation literature, these terms are often used instead of the term *Jacobian matrix*. Devices that have a conductance (Ohm’s law) representation include resistors, capacitors, diodes as well as most transistor models.

3.3 The “modified” part of “modified KCL”

The modified KCL formulation is similar to the KCL formulation, but requires at least one equation that is not a KCL equation. There are almost always non-Ohmic devices in a circuit, so additional non-KCL equations have to be added to the equation set. New vari-

ables are added to compliment these new auxiliary equations, and are generally current variables, rather than voltage variables.

The most common example of a non-Ohmic device is the independent voltage source, which imposes a predefined voltage drop across two circuit nodes, and is analogous to a Dirichlet boundary condition in a PDE problem. Because the current through the source is completely independent of this voltage drop, one cannot use an Ohm's law expression to describe it. That is, unlike most currents in the circuit, it cannot be inferred from nodal voltages. As a result, the current has to be included as a solution variable.

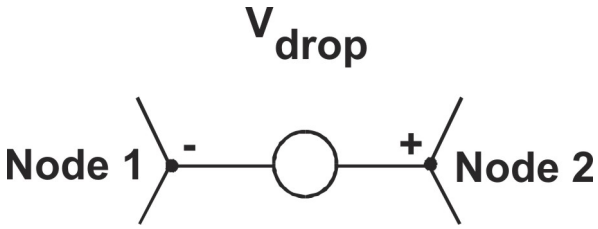

Figure 5. Independent voltage source.

Consider the voltage source illustrated in Figure 5. This source is connected to nodes 1 and 2, and so it has to be accounted for in the KCL equations for each of these nodes. The auxiliary, non-KCL equation enforces the voltage drop: $V_2 - V_1 = V_{drop}$. The source current, which is the auxilliary solution variable, is summed into the KCL equation for each of the two nodes. This enforces that the current flowing into the source from node 1 is equal to the current flowing out of the source to node 2.

3.4 A simple modified KCL example

For this example, a very simple linear circuit is illustrated in Figure 6 (For simplicity, all the example circuits in this document are variations of this circuit). The circuit is assumed not to have any time dependent elements (e.g. capacitors) and all of the devices in the circuit are linear, so it is not necessary to consider time integration or nonlinear solver issues.

In this example there two linear resistors and one independent voltage source. The three solution variables are the voltage at node 1 (V_1), the voltage at node 2 (V_2) and the current through the voltage source (I_{Vsrc}). The voltage of the ground node (node 0) is assumed to be zero volts, so V_0 is not needed as a solution variable. The ground node is never included as a variable in an analog circuit simulation. The reason for this is that voltage (electrostatic potential) is a relative quantity, and it isn't meaningful without a reference point. By convention, the ground node is always considered to be the reference, at zero volts. Without it, there would be an infinite number of solutions to the problem.

For the three solution variables there are three corresponding equations — the KCL equations for node 1 and 2, and the voltage drop equation for the voltage source. These three

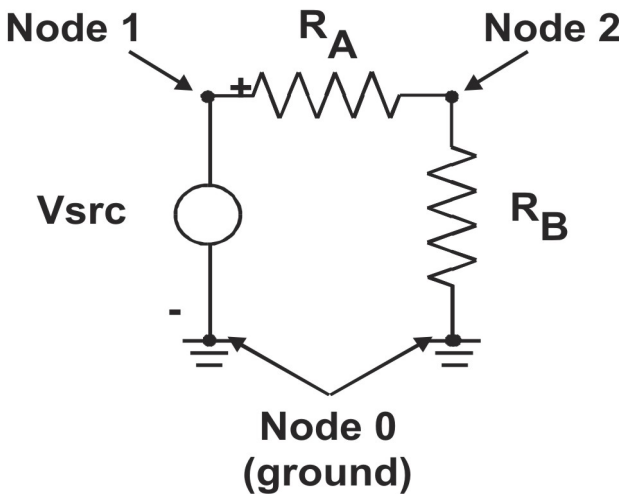

Figure 6. Simple linear, steady-state circuit.

equations are given by:

KCL equation for node 1:

$$\sum_{i=0}^{N_1} I_i = 0, \tag{6}$$

where N_1 is the number of branch currents going into/out of node 1. The current through resistor R_A is given by:

$$I_{R_A} = (V_2 - V_1)/R_A = (V_2 - V_1) \cdot G_A, \tag{7}$$

where G_A is the conductance:

$$G_A = 1/R_A. \tag{8}$$

The current through the voltage source is assumed to be whatever is required to satisfy the KCL equation. Thus, the total KCL equation is given by:

$$I_{Vsrc} - (V_2 - V_1) \cdot G_A = 0. \tag{9}$$

KCL equation for node 2:

The sum of the currents into node 2 also must sum to zero, but now instead of the voltage source, the second branch current is through R_B. So the total KCL equation for node 2 is given by:

$$(V_2 - V_1) \cdot G_A - (V_0 - V_2) \cdot G_B = 0. \tag{10}$$

Since V_0 is the ground node and assumed to be zero, this simplifies to:

$$(V_2 - V_1) \cdot G_A + V_2 \cdot G_B = 0. \tag{11}$$

Voltage drop equation:

The voltage source simply enforces that the voltage difference between node 1 and ground is held to a predefined constant: V_{drop}. Voltage source values may vary with time (as a sine wave, for example), but for this example we assume it to be constant in time. The voltage drop equation is given by:

$$V_1 - V_{drop} = 0. \tag{12}$$

Linear system:

This system of three equations can be represented by a matrix equation, which is given by:

$$\begin{bmatrix} G_A & -G_A & 1 \\ -G_A & G_A + G_B & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_{Vsrc} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ V_{drop} \end{bmatrix}, \tag{13}$$

where V_1 is the voltage at node 1, V_2 is the voltage at node 2, and I_{Vsrc} is the current through the independent voltage source. These are the three solution variables of this formulation of the problem. Most circuit problems are nonlinear and are typically solved with some form of Newton's method. Even though this example is linear, it is instructive to show how this problem would be set up for the more general nonlinear case. Recall that for Newton's method, a linear system is solved for each step of a Newton loop:

$$\mathbf{J}\Delta\mathbf{x} = -\mathbf{f} \tag{14}$$

where $\mathbf{J}$ is the Jacobian matrix for f, $\Delta\mathbf{x}$ is the update vector to be applied to the solution vector, $\mathbf{x}$, and f is the residual vector. At each Newton step, equation (14) is solved to obtain $\Delta\mathbf{x}$, and the solution vector is updated by evaluating this expression:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^{k+1} \tag{15}$$

where k is the step index. For the current example, the terms in (14, 15) are given as follows:

$$\begin{aligned}
\mathbf{f} &= \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} \text{KCL equation, node 1} \\ \text{KCL equation, node 2} \\ \text{Voltage Drop constraint equation} \end{bmatrix} \\
&= \begin{bmatrix} I_{Vsrc} - (V_2 - V_1) \cdot G_A \\ (V_2 - V_1) \cdot G_A + V_2 \cdot G_B \\ V_1 - V_{drop} \end{bmatrix} \qquad (16)
\end{aligned}$$

$$\mathbf{J} = \begin{bmatrix} \delta f_1/\delta V_1 & \delta f_1/\delta V_2 & \delta f_1/\delta I_{Vsrc} \\ \delta f_2/\delta V_1 & \delta f_2/\delta V_1 & \delta f_2/\delta I_{Vsrc} \\ \delta f_3/\delta V_1 & \delta f_3/\delta V_2 & \delta f_3/\delta I_{Vsrc} \end{bmatrix} = \begin{bmatrix} G_A & -G_A & 1 \\ -G_A & G_A + G_B & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad (17)$$

$$\mathbf{\Delta x} = \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \\ \Delta I_{Vsrc} \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} V_1 \\ V_2 \\ I_{Vsrc} \end{bmatrix}. \qquad (18)$$

Please note that the subscripts on f are meant to denote the index into the vector $\mathbf{f}$. The subscripts on V are meant to denote the nodal index for the respective voltage. Finally, the subscripts on G are meant to refer to the resistor index. In this document, resistors will always be denoted by letters (A, B, C) rather than numbers.

The KCL equations are current (I) equations, and their respective solution variables are voltage (V) variables, so most of the $\delta f/\delta x$ terms that comprise the Jacobian matrix are going to be of the general form:

$$\frac{\delta f}{\delta x} = \frac{\delta I}{\delta V} = G \qquad (19)$$

Where G is a conductance. It is only for non-Ohmic terms in $\mathbf{f}$, such as the voltage source equation, that Jacobian elements will not be in units of conductance.

Therefore, the presence of the voltage source (which necessitates a modified KCL form) changes the structure of the Jacobian matrix, $\mathbf{J}$, a great deal. There are now some non-conductance matrix contributions, which are of a fixed magnitude, 1.0. Also, the third diagonal element is zero. Both of these issues can result in the linear system being more difficult to solve, but can be addressed by scaling the problem and by matrix reordering. This example illustrates part of why circuit matrices are often ill conditioned, as for a typical digital circuit most conductances are much smaller than 1.0. A typical conductance could be around 1.0e-08 or smaller. Furthermore, the respective magnitudes of the voltage and current variables in the solution may be quite different (many orders of magnitude) and this is reflected in their associated matrix entries.

As noted, in this particular example, the two resistors are linear, so the problem is solved with a single Newton iteration. Most circuit problems, however, have nonlinear elements,

requiring multiple iterations. Some simple examples of nonlinear circuits can be found in the next section.

4 Nonlinear circuits

Most circuits contain nonlinear elements. A common example of a nonlinear device is the diode, shown in Figure 7. Like resistors and capacitors, they exist in circuit models both as stand-alone devices and also as sub-components of larger, more complex devices, such as transistors.

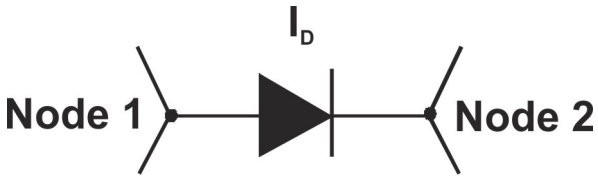

Figure 7. Diode.

Diodes are good conductors when the current flows in one direction, but poor conductors when the current flows in the opposite direction. In the simplest approximation, the current through a diode is modeled as an exponential function of the voltage drop across the two terminals. The diode current is given by:

$$I_D = I_S \left[\exp\left(\frac{V_{12}}{V_{th}} \right) - 1 \right] \tag{20}$$

I_S is a constant known as the saturation current (a typical value is 1.0×10^{-14}). $V_{12} = (V_1 - V_2)$ is the voltage drop across the diode. V_{th} is the thermal voltage and is, essentially, temperature expressed in units of eV. At room temperature it has a value of about 0.025. A plot of a typical diode current with respect to voltage is given in Figure 8. (Note that in circuit modeling, most devices are modeled using current as a function of voltage. If you read the circuit simulation literature, you will see a lot of plots of this nature). Note also, that the figure includes a breakdown current for voltages below -4.0 volts, but the example equations presented here do not include breakdown current effects.

4.1 Example: Nonlinear Circuit Problem

An example diode circuit is shown in Figure 9. This is the same circuit as was described in the previous section, with the exception that the linear resistor R_A has been replaced with a diode (which can be thought of as a nonlinear resistor). The system of equations is similar to that of the linear problem. The main difference comes from the handling of the current between node 1 and node 2.

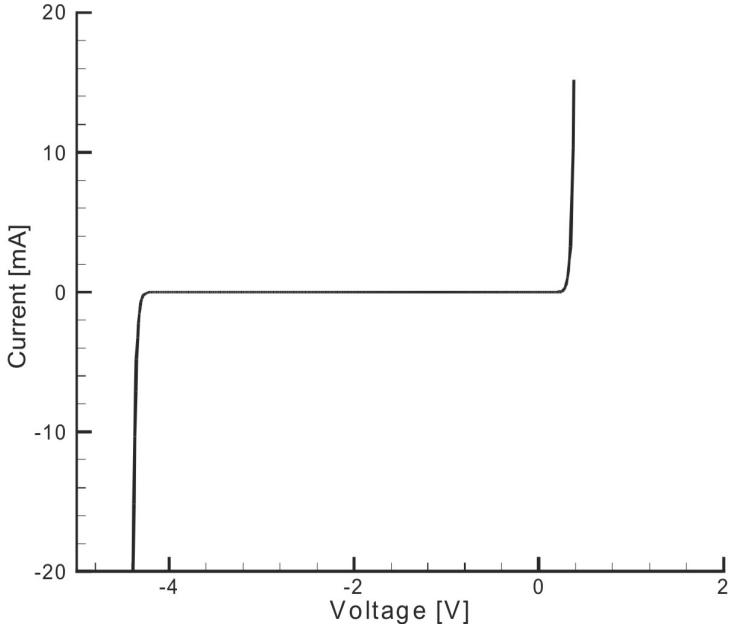

Figure 8. Diode I-V characteristic.

KCL equation for node 1:

Recall:

$$\sum_{i=0}^{N_1} I_i = 0 \tag{21}$$

As before, N_1 is the number of branch currents going into/out-of node 1. The current through the diode is given by equation (20). The current through the voltage source, like in the previous example, is assumed to be whatever it needs to be to satisfy the KCL equation. The total KCL equation for node 1 is therefore given by:

$$I_{Vsrc} + I_D = I_{Vsrc} + I_S \left[\exp\left(\frac{V_{12}}{V_{th}} \right) - 1 \right] = 0 \tag{22}$$

KCL equation for node 2:

The sum of the currents into node 2 also must sum to zero, but now instead of the voltage source, the second branch current is through R_B . So the total KCL equation for node 2 is given by:

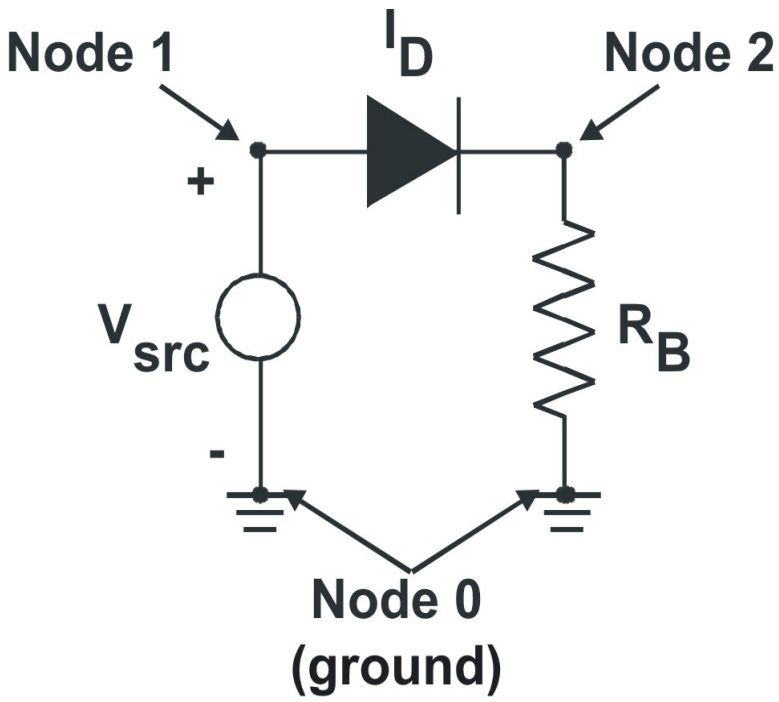

Figure 9. Diode circuit.

$$I_{R_B} - I_S \left[\exp\left(\frac{V_{12}}{V_{th}} \right) - 1 \right] = V_2 \cdot G_B - I_S \left[\exp\left(\frac{V_{12}}{V_{th}} \right) - 1 \right] = 0 \tag{23}$$

Voltage drop equation:

The voltage drop equation is the same as before (12):

$$V_1 - V_{drop} = 0 \tag{24}$$

Linear system:

The linear system to be solved at each Newton step is similar to that of the linear case and can be represented by:

$$\mathbf{J}\Delta\mathbf{x} = -\mathbf{f} \tag{25}$$

$$\begin{bmatrix} G_D & -G_D & 1 \\ -G_D & G_D + G_B & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \\ \Delta I_{Vsrc} \end{bmatrix} = - \begin{bmatrix} I_{Vsrc} + I_S \left[\exp(V_{12}/V_{th}) - 1\right] \\ V_2 \cdot G_B - I_S \left[\exp(V_{12}/V_{th}) - 1\right] \\ V_1 - V_{drop} \end{bmatrix} \tag{26}$$

However, now G_D (formerly G_A in the linear example) is a nonlinear, rather than constant, quantity. G_D is given by:

$$G_D = \frac{dI_D}{dV_1} \left(= -\frac{dI_D}{dV_2} \right) = \frac{I_S}{V_{th}} \exp\left(\frac{V_{12}}{V_{th}}\right) = \frac{I_S}{V_{th}} \exp\left(\frac{V_1 - V_2}{V_{th}}\right) \tag{27}$$

The system of nonlinear equations is solved in **Xyce** using Newton's method. Generally, circuits including lots of exponential I-V relationships require some enhancement to Newton's method, such as a line search. A discussion of the various nonlinear solver options in **Xyce** is beyond the scope of this document (see [8] for the available options and their use).

One nonlinear solution method is worth special attention: voltage limiting. This approach is somewhat unique to circuit simulation and is covered in the next section. In many circuit codes (almost any code based on SPICE), voltage limiting is the only Newton solver enhancement.

4.2 Voltage Limiting

Voltage limiting is a method for enhancing the nonlinear solve portion of a circuit simulation. The idea behind it is to prevent voltage drops in some of the semiconductor devices from changing too much from one Newton step to the next. It is the only nonlinear solver enhancement available in Spice3f5, and it is hardcoded to always be invoked. It appears to have been used in some of the earliest circuit simulators [10]. For a **Xyce** developer, it is important to understand voltage limiting, as it is an unusual technique with a number of consequences. In practice, it has proven to be a very effective method for obtaining difficult solutions, but it is incompatible with most conventional nonlinear solver enhancements. As will be shown in this section, there are three reasons for this incompatibility.

1. Voltage limiting directly changes the right hand side vector used by the Newton iteration, so that it contains more than just $\mathbf{f}$. It changes the set of equations to be solved.

2. Voltage limiting results in a different Newton update to the solution vector, but it does so in an inconsistent way. As a result, it is difficult to reproduce it using a technique applied to the entire solution vector. For example, using $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \cdot \Delta\mathbf{x}$ and varying the magnitude of the scalar α will not yield the same result.

3. The limiting relies on the previous Newton step size, meaning that it is a function of the path taken to the current solution. This means that any technique using backtracking would be subject to hysteresis.

As **Xyce** and Spice3f5 use different formulations for the Newton solve, there are algebraic differences between the two codes in voltage limiting implementation. In this section, the Spice3f5 implementation of voltage limiting will be described first, followed by a description of the implementation in **Xyce**.

Voltage Limiting in Spice3f5

In Spice3f5, the solver solves directly for the new value of the solution at each nonlinear step, rather than solving for the update, which is more standard for nonlinear solvers. Most traditional nonlinear solvers will solve this nonlinear system:

$$\mathbf{J}\Delta\mathbf{x}^{k+1} = -\mathbf{f} \tag{28}$$

$$\Delta\mathbf{x}^{k+1} = \mathbf{x}^{k+1} - \mathbf{x}^{k} \tag{29}$$

The index, k, is the Newton iteration step number. In contrast, the Spice3f5 nonlinear iteration is accomplished by solving this equivalent linear system:

$$\mathbf{J}\mathbf{x}^{k+1} = -\mathbf{f} + \mathbf{J}\mathbf{x}^{k} \tag{30}$$

Recall that most of the elements of are nodal voltages, most of the elements of $\mathbf{f}$ are currents and most of the elements of $\mathbf{J}$ are in units of conductance.

Unfortunately, the Spice3f5 approach to Newton's method means that it is impossible (or at least difficult) to apply traditional nonlinear solver libraries (such as NOX [3]). The right hand side of the Equation 30 can no longer be assumed to be $-\mathbf{f}$, as it contains the $J\mathbf{x}^{k}$ terms.

In Spice3f5, to implement voltage limiting, an analysis is performed at the beginning of each Newton step, to determine if the previous step resulted in any junction voltage changes that were too large. (A junction voltage is the difference between two connected nodal voltages. An example would be a voltage drop across a diode) In the event that some of them were too large, portions of $\mathbf{x}_i$ are replaced with values that are acceptable to the limiting scheme. Then *after* $\mathbf{x}_i$ has been modified, the calculation proceeds as though this modified x_i is the correct one. $\mathbf{J}$ and $\mathbf{f}$ are both calculated using this modified version of $\mathbf{x}_i$. In a sense, the code goes into denial.

It should be noted that this is slightly more complicated than it may first appear, because all of this is done in terms of *junction voltages* (voltage drops between nodes), not *nodal voltages*. Nodal voltages are what actually exist as distinct elements of the solution vector $\mathbf{x}_i$, but junction voltages are what most devices actually care about. If we consider the diode, again, recall the expressions for diode current and the Jacobian contribution are:

$$I_D = I_S \left[\exp\left(\frac{V_{12}}{V_{th}}\right) - 1 \right] \tag{31}$$

$$G_D = \frac{I_S}{V_{th}} \exp\left(\frac{V_{12}}{V_{th}}\right) \tag{32}$$

Both I_D and G_D are functions of $V_{12} = V_1 - V_2$, which is a junction voltage. When a circuit code calculates the contributions of a diode to $\mathbf{J}$ and $\mathbf{f}$, it first obtains V_1 and V_2 from the solution vector, and then immediately obtains V_{12}. From that point onward in the calculation, V_1 and V_2 are ignored and everything is calculated as a function of V_{12}. This is typical in all Spice3f5-style analytical device models.

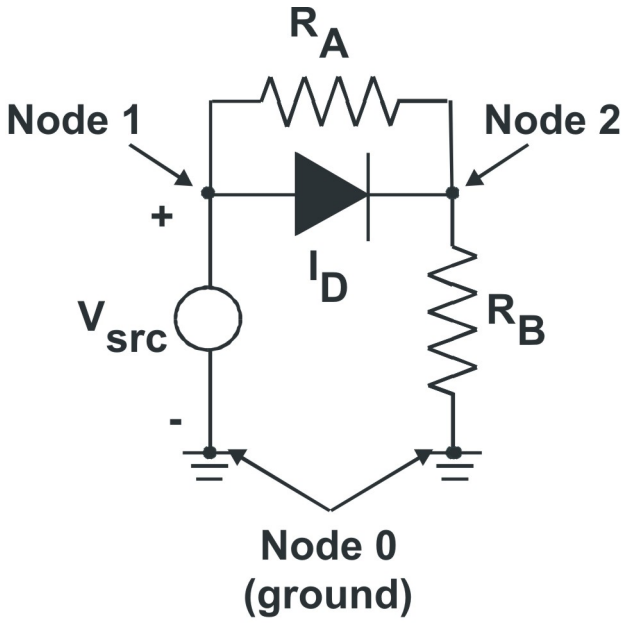

Figure 10. Diode circuit with resistor in parallel.

In practice, the $\mathbf{J}\mathbf{x}^k$ term on the right hand side of (30) is not handled by doing a formal matrix-vector multiply, so it is easy to implement a junction voltage based limiting scheme. To illustrate this, consider the diode example from Figure 10. This is the same diode circuit as from Figure 9, only an extra nonlinear resistor, R_A, has been added in parallel with the

diode. (The extra resistor has been added in order to illustrate one of the more subtle aspects of voltage limiting.) The number of nodes is the same as before, and the number of solution variables is also the same, but the number of branch currents to be considered for the KCL equations of node 1 and 2 has increased.

The difference between the nonlinear resistor R_A, and the linear resistor R_B, is that for R_A: $I_A \neq G_A \cdot V_{12}$, but for $R_B : I_B = G_B \cdot V_{20}$. To solve this problem in Spice3f5, the linear system from equation (30) to be solved at each Newton step (without voltage limiting) is:

$$\begin{aligned}\mathbf{J}\mathbf{x}^{k+1} = & - \begin{bmatrix} I_{Vsrc}^k + I_S\left[\exp\left(V_{12}^k/V_{th}\right) - 1\right] + I_A \\ G_B \cdot V_2^k - I_S\left[\exp\left(V_{12}^k/V_{th}\right) - 1\right] - I_A \\ V_1^k - V_{drop} \end{bmatrix} \\ & + \begin{bmatrix} G_D + G_A & -G_D - G_A & 1 \\ -G_D - G_A & G_D + G_A + G_B & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^k \\ V_2^k \\ I_{Vsrc}^k \end{bmatrix} \end{aligned} \tag{33}$$

where:

$$\mathbf{J} = \begin{bmatrix} G_D + G_A & -G_D - G_A & 1 \\ -G_D - G_A & G_D + G_A + G_B & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{34}$$

and:

$$\mathbf{x}^{k+1} = \begin{bmatrix} V_1^{k+1} \\ V_2^{k+1} \\ I_{Vsrc}^{k+1} \end{bmatrix} \tag{35}$$

Note that the Jacobian elements are all evaluated in terms of old (iteration k) variables. That is:

$$J = J(\mathbf{x}^k) \tag{36}$$

Recall that G_A and G_B are constant for the linear resistor. This next linear system is slightly more representative of the way the problem is actually implemented in the code:

$$\mathbf{J}\mathbf{x}^{k+1} =$$

$$- \begin{bmatrix} I_{Vsrc}^k + I_S\left[\exp\left(V_{12}^k/V_{th}\right) - 1\right] + I_A - G_D \cdot V_{12}^k - G_A \cdot V_{12}^k - I_{Vsrc}^k \\ G_B \cdot V_2^k - I_S\left[\exp\left(V_{12}^k/V_{th}\right) - 1\right] - I_A + G_D \cdot V_{12}^k + G_A \cdot V_{12}^k - G_B \cdot V_2^k \\ V_1^k - V_{drop} - V_1^k \end{bmatrix} \tag{37}$$

The right hand side terms have been combined. Rewriting, in terms of junction voltages, and after canceling terms:

$$\mathbf{J}\mathbf{x}^{k+1} = -\begin{bmatrix} I_S\left[\exp\left(V_{12}^k/V_{th}\right)-1\right] + I_A - G_D\cdot V_{12}^k - G_A\cdot V_{12}^k \\ -I_S\left[\exp\left(V_{12}^k/V_{th}\right)-1\right] - I_A + G_D\cdot V_{12}^k + G_A\cdot V_{12}^k \\ -V_{drop} \end{bmatrix} \tag{38}$$

Note that the linear resistor contributions have vanished from the right hand side, but the nonlinear ones remain. In practice, there are no linear resistor contributions to the right hand side in Spice3f5 implementation.

In most codes, the matrix equation is set up on a device-by-device basis. The simulation program stores all of the devices for a given circuit and loops through them in the process of setting up the matrix and right hand side vector by way of summation. In this example, if the order that the four devices appear in the list is: diode, voltage source, R_A, and R_B, then the linear system would be at each stage:

After the diode load:

$$\begin{bmatrix} G_D & -G_D & 0 \\ -G_D & G_D & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^{k+1} \\ V_2^{k+1} \\ I_{Vsrc}^{k+1} \end{bmatrix} = -\begin{bmatrix} I_S\left[\exp\left(V_{12}^k/V_{th}\right)-1\right] - G_D\cdot V_{12}^k \\ -I_S\left[\exp\left(V_{12}^k/V_{th}\right)-1\right] + G_D\cdot V_{12}^k \\ 0 \end{bmatrix} \tag{39}$$

After the voltage source load:

$$\begin{bmatrix} G_D & -G_D & 1 \\ -G_D & G_D & 0 \\ 1 & 0 & 0 \end{bmatrix} \mathbf{x}^{k+1} = -\begin{bmatrix} I_S\left[\exp\left(V_{12}^k/V_{th}\right)-1\right] - G_D\cdot V_{12}^k \\ -I_S\left[\exp\left(V_{12}^k/V_{th}\right)-1\right] + G_D\cdot V_{12}^k \\ -V_{drop} \end{bmatrix} \tag{40}$$

After the nonlinear resistor, R_A load:

$$\begin{bmatrix} G_D + G_A & -G_D - G_A & 1 \\ -G_D - G_A & G_D + G_A & 0 \\ 1 & 0 & 0 \end{bmatrix} \mathbf{x}^{k+1} = -\begin{bmatrix} I_S\left[\exp\left(V_{12}^k/V_{th}\right)-1\right] - G_D\cdot V_{12}^k + I_A - G_A\cdot V_{12}^k \\ -I_S\left[\exp\left(V_{12}^k/V_{th}\right)-1\right] + G_D\cdot V_{12}^k - I_A + G_A\cdot V_{12}^k \\ -V_{drop} \end{bmatrix} \tag{41}$$

And, finally, after the resistor, R_B, load:

$$\begin{bmatrix} G_D + G_A & -G_D - G_A & 1 \\ -G_D - G_A & G_D + G_A + G_B & 0 \\ 1 & 0 & 0 \end{bmatrix} \mathbf{x}^{k+1} =$$

$$-\begin{bmatrix} I_S\left[\exp\left(V_{12}^k/V_{th}\right)-1\right]-G_D\cdot V_{12}^k+I_A-G_A\cdot V_{12}^k \\ -I_S\left[\exp\left(V_{12}^k/V_{th}\right)-1\right]+G_D\cdot V_{12}^k-I_A+G_A\cdot V_{12}^k \\ -V_{drop} \end{bmatrix} \tag{42}$$

The main reason for describing the device-by-device load is to illustrate that the load calculations for each device are largely independent of each other.

As noted, voltage limiting involves the code checking the old junction voltages, such as V_{12}^{old}, and replacing them with different values if necessary. The voltage limiting procedure for a single device, at each nonlinear step is illustrated in the flow chart in Figure 11.

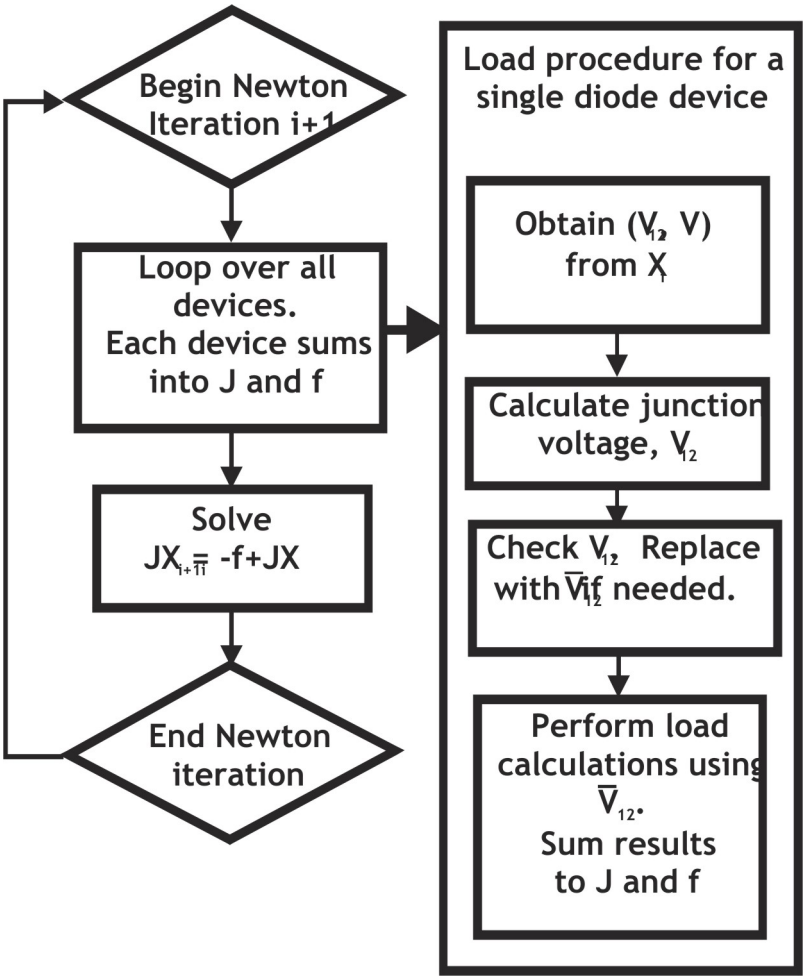

Figure 11. Voltage limiting flowchart.

As the load procedure is done on a device-by-device basis, the decision to replace an old junction voltage is also done on a device-by-device basis. Not all devices have voltage limiter functions and those that do generally do not all have the same voltage limiter functions.

For example, consider the current example in which voltage limiting is applied in the diode but not in R_A. The resulting linear equation is:

$$\begin{bmatrix} G_D+G_A & -G_D-G_A & 1 \\ -G_D-G_A & G_D+G_A+G_B & 0 \\ 1 & 0 & 0 \end{bmatrix} \mathbf{x}^{k+1} = - \begin{bmatrix} I_S\left[\exp\left(\hat{V}_{12}^k/V_{th}\right)-1\right] - G_D\cdot\hat{V}_{12}^k + I_A - G_A\cdot V_{12}^k \\ -I_S\left[\exp\left(\hat{V}_{12}^k/V_{th}\right)-1\right] + G_D\cdot\hat{V}_{12}^k - I_A + G_A\cdot V_{12}^k \\ -V_{drop} \end{bmatrix} \tag{43}$$

Note that for all the terms associated with the diode, V_{12}^k has been replaced by $\hat{V}_{12}^k$. As such, the voltage drop used in the diode calculation is different than that used in the R_A calculation, even though they are attached to the same nodes of the circuit. It does not modify the old (iteration k) solution vector in a consistent manner. It was to demonstrate this issue that R_A was added to this example circuit. Voltage limiting is very similar to conventional nonlinear enhancements, such as constraint backtracking. However, a constraint backtracking scheme (which is applied outside of the load procedure, and could not easily include this inconsistency) would not be able to exactly reproduce it. Of course, it may be preferable to avoid such an inconsistency.

Generally, the voltage-limiting technique is used for semiconductor device models only, such as diodes and transistors (e.g., BJTs and MOSFETs), which may be highly nonlinear. Voltage limiting rules are designed to prevent junction voltages from changing too much between iterations, and the extent to which they are allowed to change is usually a function of the device's I-V characteristic. In general, if the device is in a regime in which $\frac{dI}{dV}$ is large, the limits are more restrictive than if the device is in a regime in which $\frac{dI}{dV}$ is small. As such, the limit imposed on V_{12}^{k+1} is a function of V_{12}^k and because the point is to constrain the change from Newton step to Newton step, it is also a function of V_{12}^{k-1}. This points to another important issue. This particular type of constraint is dependent not just on the current values in the solution vector, but the path taken by the solver to get there. As a result, there is hysteresis in the solution technique.

Voltage Limiting in **Xyce**

As noted, for a variety of reasons **Xyce** uses a more traditional approach to obtain the solution to the nonlinear problem $f(\mathbf{x}) = 0$:

$$\mathbf{J}\Delta\mathbf{x}^{k+1} = -\mathbf{f}, \text{ where} \tag{44}$$

$$\Delta\mathbf{x}^{k+1} = \mathbf{x}^{k+1} - \mathbf{x}^k. \tag{45}$$

Initially, voltage limiting was not a planned feature for **Xyce**, in part because the nonlinear solver was designed to use the traditional Newton iteration, as described by Equation 44. Once the value of voltage limiting became apparent, it was necessary to reformulate Spice3f5 implementation to work in **Xyce**.

For the nonlinear iteration k, the original "unlimited" solution vector is given by $\mathbf{x}^k$, the intermediate limited solution vector is given by $\hat{\mathbf{x}}^{k+1}$, and the final solution vector at the end of the iteration is given by $\mathbf{x}^{k+1}$. Thus, for a Newton step that includes voltage limiting, the total change from the beginning of the step to the end is:

$$\Delta\mathbf{x}_{total}^{k+1} = \Delta\mathbf{x}_{newton}^{k+1} + \Delta\hat{\mathbf{x}}^{k+1} \tag{46}$$

where:

$$\Delta\hat{\mathbf{x}}^{k+1} = \hat{\mathbf{x}}^{k+1} - \mathbf{x}^{k} \tag{47}$$

and:

$$\Delta\mathbf{x}_{newton}^{k+1} = \mathbf{x}^{k+1} - \hat{\mathbf{x}}^{k+1} \tag{48}$$

$\Delta\hat{\mathbf{x}}^{k+1}$ represents the change in the solution due to voltage limiting, $\Delta\mathbf{x}_{newton}^{k+1}$ represents the change due to the solution of the matrix equation, and $\Delta\mathbf{x}_{total}^{k+1}$ is the total change over the course of the Newton step. The linear equation to be solved is now:

$$\mathbf{J}\Delta\mathbf{x}_{total}^{k+1} = -\mathbf{f} + \mathbf{J}\Delta\hat{\mathbf{x}}^{k+1} \tag{49}$$

This equation has been obtained by adding $\mathbf{J}\Delta\hat{\mathbf{x}}^{k+1}$ to both sides of the original Newton step equation.

The calculations performed at each Newton step are very similar using this approach as they were for the Spice3f5 case. The **Xyce** version of equation (43) is given by:

$$\mathbf{J}\Delta\mathbf{x}_{total}^{k+1} = -\begin{bmatrix} I_S\left[\exp\left(\hat{V}_{12}^{k+1}/V_{th}\right)-1\right] + I_A - G_D\cdot\Delta\hat{V}_{12}^{k+1} \\ -I_S\left[\exp\left(\hat{V}_{12}^{k+1}/V_{th}\right)-1\right] - I_A + G_D\cdot\Delta\hat{V}_{12}^{k+1} \\ \mathbf{V}_1^{k+1} - V_{drop} \end{bmatrix} \tag{50}$$

Or, alternatively:

$$\mathbf{J}\Delta\mathbf{x}_{total}^{k+1} = -\begin{bmatrix} I_S\left[\exp\left(\hat{V}_{12}^{k+1}/V_{th}\right)-1\right]+I_A \\ -I_S\left[\exp\left(\hat{V}_{12}^{k+1}/V_{th}\right)-1\right]-I_A \\ V_1^{k+1}-V_{drop} \end{bmatrix} + \begin{bmatrix} G_D\cdot\hat{V}_{12}^{k+1} \\ -G_D\cdot\hat{V}_{12}^{k+1} \\ 0 \end{bmatrix} \tag{51}$$

$$\begin{aligned} \mathbf{J}\Delta\mathbf{x}_{total}^{k+1} = \quad & - \begin{bmatrix} I_S\left[\exp\left(\hat{V}_{12}^{k+1}/V_{th}\right)-1\right]+I_A \\ -I_S\left[\exp\left(\hat{V}_{12}^{k+1}/V_{th}\right)-1\right]-I_A \\ V_1^{k+1}-V_{drop} \end{bmatrix} \\ & + \begin{bmatrix} G_D & -G_D & 0 \\ -G_D & G_D & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\hat{V}_1^{k+1} \\ \Delta\hat{V}_2^{k+1} \\ \Delta\hat{I}_{Vsrc}^{k+1} \end{bmatrix} \end{aligned} \tag{52}$$

One nice feature of this formulation is that as the Newton algorithm approaches convergence, the values for $\Delta\mathbf{x}_{total}^{k+1}$ become much smaller. By solving for $\Delta\mathbf{x}^{k+1}$ rather than $\mathbf{x}^{k+1}$, it is easier to resolve the small changes in the solution that occur during the final iterations. Also, as most nonlinear solver libraries and algorithms are designed for this ($\Delta\mathbf{x}$) approach, it is much easier to take advantage of them. Finally, the final ($\mathbf{J}\Delta\hat{\mathbf{x}}^{k+1}$) term on the right hand side of equation (52) can easily be stored in a separate vector than the first ($-\mathbf{f}$) term, so algorithms depending upon f are impacted less than in the other case.

Additional Notes

Xyce has a large number of different nonlinear solver options, including damped Newton, modified Newton, inexact-Newton, constraint backtracking, and gradient-based methods. A complete list of options, and a guide to usage is contained in [8]. In general, most nonlinear solver strategies are incompatible with voltage limiting, either because of the unorthodox right hand side vector, or because of the hysteresis introduced by the limiters. Some future work may include finding ways to combine the effect of the limiters with some of the other methods. The voltage limiter technique has been one of the most effective techniques, especially for semiconductor circuits.

5 Time Dependent Circuits

In practice, most circuits contain a number of time dependent elements, and many of those elements (capacitors and inductors) are described by ordinary differential equations (ODEs) that include time derivative terms. For example, the current in a linear capacitor is given by:

$$I_C = \frac{dq}{dt} \tag{53}$$

Capacitors are particularly ubiquitous, as not only are they usually present as distinct devices, but they also appear as subcomponents in every semiconductor device model. While the inclusion of such devices requires that the mathematical formulation include ODEs, the overall formulation is still based upon modified nodal analysis, and as such, the set of equations to be solved contains a number of purely algebraic equations. These include the voltage drop equation for a voltage source, or any KCL equation that includes only resistor currents. The set of equations, therefore, is a set of differential-algebraic equations (DAEs), where those that are purely algebraic are considered to be the constraints.

There is a lot of literature devoted to describing DAEs and methods for their solution [5, 4]. For the most part, that material will not be described here as it is beyond the scope of this document. Most systems of equations resulting from circuit theory can be cast as DAEs of index one, and fortunately the techniques for solving DAEs of index one are fairly well understood. It is possible, particularly in circuits containing operational amplifiers, to obtain DAEs of much higher index [5], but at the moment such circuits are not possible to simulate in **Xyce**, so they will not be considered here. **Xyce** and Spice3f5 both use a condensed form of the circuit equations that will be described in the third subsection of this section. This condensed form appears to be a standard technique for circuit simulation, and has some obvious advantages, but for the most part this condensed form has not been described in the literature. In practice, it appears to work reasonably well, but the numerical implications (stability, accuracy, etc.) of using such a form are not entirely clear at this point.

5.1 Traditional Index-1 DAE Formulation for the Linear Case

Differential Algebraic Equations (DAEs) generally have the form:

$$\mathbf{f}\left(\mathbf{x}, \frac{d\mathbf{x}}{dt}, t\right) = 0 \tag{54}$$

For the linear case, this is often presented in the form of a matrix equation:

$$\mathbf{f} = \mathbf{A}\mathbf{x} + \mathbf{B}\frac{d\mathbf{x}}{dt} + \mathbf{r}(t) = 0 \tag{55}$$

In equation (55), $\mathbf{A}$ and $\mathbf{B}$ are matrices and $\mathbf{r}(t)$ is a source term. This formalism can easily be applied to the linear circuit that is presented in Figure 12. This circuit is the same as all of the previous example circuits, except that now a capacitor sits between nodes 1 and 2. The capacitor, C_A, is a time dependent device, and has a current defined to be:

$$I_{C_A} = \frac{dq}{dt} \tag{56}$$

where q is the charge stored by the capacitor, which is defined to be:

$$q = C_A \cdot V_{12} \tag{57}$$

where C_A is the capacitance and $V_{12} = V_1 - V_2$ is the voltage drop across the capacitor. For the linear case (where C_A is constant), equation (56) can be simplified to be:

$$I_{C_A} = C_A \frac{dV_{12}}{dt} \tag{58}$$

For the linear case, the linear system consists of three equations and three unknowns. The system is very similar to that described by equations (6) through (13), except that a capacitor has replaced one of the resistors. Thus, the three solution variables are given by: (V_1, V_2, I_{Vsrc}) The three equations, like before, consist of KCL equations for nodes 1 and 2, plus a voltage drop equation for the independent source.

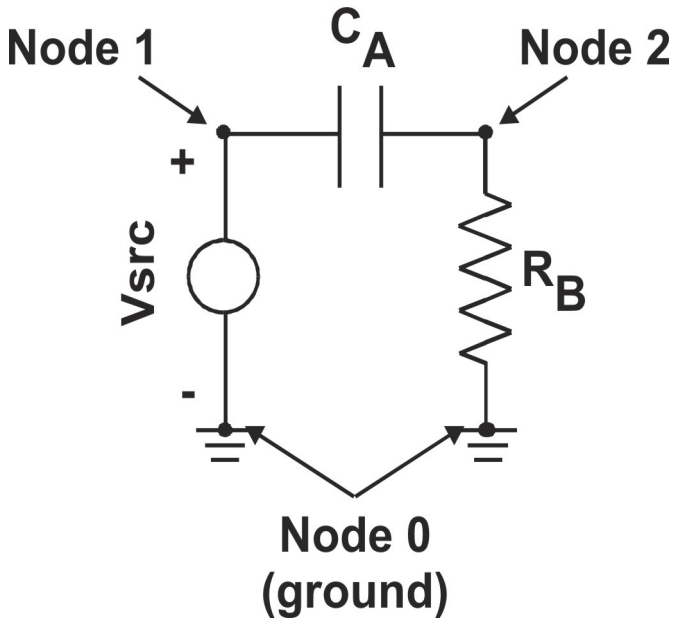

Figure 12. Linear time dependent circuit.

KCL equation for node 1:

$$I_{Vsrc} - C_A \cdot \frac{d}{dt}\left(V_{21}^{i+1}\right) = I_{Vsrc} - C_A \cdot \frac{d}{dt}\left(V_2^{i+1} - V_1^{i+1}\right) = 0 \tag{59}$$

KCL equation for node 2:

$$C_A \cdot \frac{d}{dt}\left(V_{21}^{i+1}\right) + V_2^{i+1} \cdot G_B = C_A \cdot \frac{d}{dt}\left(V_2^{i+1} - V_1^{i+1}\right) + V_2^{i+1} \cdot G_B = 0 \tag{60}$$

Voltage drop equation:

$$V_1^{i+1} - V_{drop} = 0 \tag{61}$$

Full system:

This set of equations can be rewritten in the form of equation (55).

$$\mathbf{f}\left(\mathbf{x}, \frac{d\mathbf{x}}{dt}, t\right) = 0 =$$

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & G_B & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^{i+1} \\ V_2^{i+1} \\ I_{Vsrc}^{i+1} \end{bmatrix} + \begin{bmatrix} C_A & -C_A & 0 \\ -C_A & C_A & 0 \\ 0 & 0 & 0 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} V_1^{i+1} \\ V_2^{i+1} \\ I_{Vsrc}^{i+1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -V_{drop} \end{bmatrix} \tag{62}$$

The addition of the capacitor requires that dV_{12}/dt be evaluated. Typically, this is done using a backward differentiation formula (BDF). For the purposes of this example we will use the backward Euler method.

$$\frac{dV_{12}}{dt} = \alpha\left(V_{12}^{i+1} - V_{12}^{i}\right) \tag{63}$$

Where $\alpha = 1/h$, and h is the time step size. The factor, α, is the leading coefficient of the BDF formula, and will have a different form depending on the choice of BDF. Incorporating equation (63) into equations (59-61) results in the KCL equations being recast as differential algebraic equations:

KCL equation for node 1:

$$I_{Vsrc}^{i+1} - \alpha \cdot C_A \cdot \left[\left(V_2^{i+1} - V_1^{i+1}\right) - \left(V_2^{i} - V_1^{i}\right)\right] = 0 \tag{64}$$

KCL equation for node 2:

$$\alpha \cdot C_A \cdot \left[\left(V_2^{i+1} - V_1^{i+1}\right) - \left(V_2^{i} - V_1^{i}\right)\right] + V_2^{i+1} \cdot G_B = 0 \tag{65}$$

Voltage drop equation:

$$V_1^{i+1} - V_{drop} = 0 \tag{66}$$

Full system:

The matrix equation then becomes (for the backward Euler case):

$$\mathbf{f}\left(\mathbf{x}, \frac{d\mathbf{x}}{dt}, t\right) = [\mathbf{A} + \alpha\mathbf{B}]\,\mathbf{x}^{i+1} + \alpha\mathbf{B}\mathbf{x}^{i} + \mathbf{r}(t) = 0 \tag{67}$$

$$\left[\begin{bmatrix} 0 & 0 & 1 \\ 0 & G_B & 0 \\ 1 & 0 & 0 \end{bmatrix} + \alpha \begin{bmatrix} C_A & -C_A & 0 \\ -C_A & C_A & 0 \\ 0 & 0 & 0 \end{bmatrix}\right] \begin{bmatrix} V_1^{i+1} \\ V_2^{i+1} \\ I_{Vsrc}^{i+1} \end{bmatrix} + \alpha \begin{bmatrix} C_A & -C_A & 0 \\ -C_A & C_A & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^{i} \\ V_2^{i} \\ I_{Vsrc}^{i} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -V_{drop} \end{bmatrix} = 0 \tag{68}$$

This is a linear problem, so the Jacobian matrix is given by:

$$\mathbf{J} = [\mathbf{A} + \alpha\mathbf{B}] = \left[\begin{bmatrix} 0 & 0 & 1 \\ 0 & G_B & 0 \\ 1 & 0 & 0 \end{bmatrix} + \alpha \begin{bmatrix} C_A & -C_A & 0 \\ -C_A & C_A & 0 \\ 0 & 0 & 0 \end{bmatrix}\right] \tag{69}$$

This example formulation is a DAE system of index zero. Recall that Petzold [5] defines the index of a DAE to be, the minimum number of times that all or part of (54) must be differentiated with respect to t in order to determine $d\mathbf{x}/dt$ as a continuous function of $(\mathbf{x}, t)$, is the index of the DAE (54). Also, she defines a DAE with an index of 0 to be equivalent to a system of ODEs. In this example, the two KCL equations are differential equations and the voltage drop equation from the voltage source is a constraint equation. However, the one constraint equation can be removed easily, without performing any differentiations. The voltage source sets the first node to a particular voltage, so including V_1 and I_{Vsrc} in the set of equations isn't necessary. The problem can thus be represented as:

$$f = C_A \cdot \frac{d}{dt}\left(V_2^{i+1} - V_{Vsrc}\right) + V_2^{i+1} \cdot G_B = 0 \tag{70}$$

or:

$$f = C_A \cdot \frac{dV_2^{i+1}}{dt} + V_2^{i+1} \cdot G_B = 0 \tag{71}$$

Equation (71) is a single implicit ODE. This conversion to an ODE, which removes the algebraic constraints, results in the need for an initial condition on V_2. It should be noted that most circuit problems do not result in a DAE of index zero. Instead, the systems are typically of index one, and some examples of higher index systems will be given in following sections. For linear circuits, the DAE formulation just described (before the removal of the voltage source equation) matches that of **Xyce**. In the next section, the most obvious way of handling the nonlinear case will be addressed.

5.2 Traditional Index-1 DAE Formulation for the Nonlinear Case

The linear example can be extended to the nonlinear case by assuming the capacitor is a nonlinear capacitor, and that $C^{i+1} = f(V_1^{i+1}, V_2^{i+1})$. The simplification of equation (58) is no longer valid, and $dq/dt \neq CdV/dt$, so it is necessary to keep track of q as a variable with respect to time. From a physical standpoint, q must be calculated as a function of time in order to enforce charge conservation. The easiest way to accomplish this is to add q to the system of solution variables, $\mathbf{x}$, and add the equation defining q as a function of V to the system of equations, $\mathbf{f}$. The DAE form is extended for the nonlinear case:

$$\mathbf{f} = \mathbf{A}\mathbf{x} + \mathbf{B}\frac{d\mathbf{x}}{dt} + \mathbf{p}(\mathbf{x}) + \mathbf{r}(t) = 0 \tag{72}$$

The term $\mathbf{p}(\mathbf{x})$ represents the nonlinear terms. For the purposes of this example, assume that the charge on C_A, q, is given by:

$$q = C_0\left(1 + V_{21}^2\right) = C_0\left[1 + (V_2 - V_1)^2\right] \tag{73}$$

C_0 is a constant prefactor and not representative of the entire capacitance. The capacitance for C_A is given by:

$$C_A = \frac{q}{V_{21}} = \frac{C_0 + C_0 V_{21}^2}{V_{21}} = \frac{C_0}{V_{21}} + C_0 \cdot V_{21} \tag{74}$$

Equation (73) is added to $\mathbf{f}$. The full system of equations is now:

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & G_B & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_1^{i+1} \\ V_2^{i+1} \\ I_{V_{src}}^{i+1} \\ q^{i+1} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} V_1^{i+1} \\ V_2^{i+1} \\ I_{V_{src}}^{i+1} \\ q^{i+1} \end{bmatrix}$$

$$+\begin{bmatrix} 0 \\ 0 \\ 0 \\ -C_0\left[1+(V_2-V_1)^2\right] \end{bmatrix}+\begin{bmatrix} 0 \\ 0 \\ -V_{drop} \\ 0 \end{bmatrix}=0 \tag{75}$$

If, like before, we assume backward Euler ($dq/dt = \alpha(q^{i+1} - q^i)$), the set of equations can be rewritten as:

$$[\mathbf{A}+\alpha\mathbf{B}]\,\mathbf{x}^{i+1}+\beta\mathbf{B}+\mathbf{p}\left(\mathbf{x}^{i+1}\right)+\mathbf{r}(t)=0 \tag{76}$$

where:

$$\mathbf{A}=\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & G_B & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{77}$$

$$\mathbf{B}=\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{78}$$

$$\beta=-\alpha\begin{bmatrix} V_1^i \\ V_2^i \\ I_{Vsrc}^i \\ q^i \end{bmatrix} \tag{79}$$

$$\mathbf{p}(\mathbf{x})=\begin{bmatrix} 0 \\ 0 \\ 0 \\ -C_0\left[1+(V_2-V_1)^2\right] \end{bmatrix} \tag{80}$$

$$\mathbf{r}(\mathbf{t})=\begin{bmatrix} 0 \\ 0 \\ -V_{drop} \\ 0 \end{bmatrix} \tag{81}$$

Unlike in the linear case, this set of equations is a set of DAEs of index one. The conversion of this DAE to an index zero DAE will be described in the next section. It should be noted that if a different BDF formula (other than backward Euler) were used to obtain time derivatives, the only thing that would change in the above set of equations would be

equation (79), the equation for β. For the nonlinear case, the Jacobian matrix is defined as:

$$\mathbf{J} = \mathbf{A} + \alpha\mathbf{B} + \frac{\delta\mathbf{p}}{\delta\mathbf{x}^{i+1}} \tag{82}$$

In the current example, the last term of equation (82) is given by:

$$\frac{\delta\mathbf{p}}{\delta\mathbf{x}^{i+1}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2C_0(V_2 - V_1) & 2C_0(V_1 - V_2) & 0 & 0 \end{bmatrix} \tag{83}$$

The final form for the Jacobian in this example is:

$$\mathbf{J} = \begin{bmatrix} 0 & 0 & 1 & \alpha \\ 0 & G_B & 0 & -\alpha \\ 1 & 0 & 0 & 0 \\ 2C_0(V_2 - V_1) & 2C_0(V_1 - V_2) & 0 & 1 \end{bmatrix} \tag{84}$$

The formulation described here is not the one used in **Xyce** or Spice3f5, but it is related. The time integration package used within **Xyce** was originally designed with the intention of using this formulation. For a variety of reasons (explained in the next section), a different, more compact formulation is used instead. The consequences of using this compact scheme, in terms of its effect on error analysis, step-size control, stability, etc., are not clear at this point.

5.3 Condensed DAE formulation in **Xyce**: State Variables

The formulation used in **Xyce** (and Spice3f5) is more compact than that of the previous section, in that is not considered as a member of the solution vector, $\mathbf{x}$. Removing q requires that the charge equation (73) be combined into some of the other equations of the system. If one takes the time derivative of the charge equation, and then substitutes the result into the appropriate KCL equations, the resulting set of equations is:

$$\mathbf{f}\left(\mathbf{x}, \frac{d\mathbf{x}}{dt}, t\right) = 0 =$$

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & G_B & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^{i+1} \\ V_2^{i+1} \\ I_{Vsrc}^{i+1} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} C_0\left[1 + (V_2 - V_1)^2\right] \\ -C_0\left[1 + (V_2 - V_1)^2\right] \\ 0 \end{bmatrix}^{i+1} + \begin{bmatrix} 0 \\ 0 \\ -V_{drop} \end{bmatrix} \tag{85}$$

This procedure appears to have the effect of reducing the index from one to zero, because now equation (85) is in a similar form to that of equation (62) before the application of the BDF. It is still a DAE, however, and not an ODE system. To reduce it further to an ODE system, one should follow the same procedure as outlined in equations (70) and (71), and remove the constraint equation for the voltage source.

Generally the lower the index of a DAE, the easier it is to solve, although there can be unintended consequences. Another (perhaps more important) benefit of this condensation is that the size of the linear system has been reduced. As will be explained later, this type of equation elimination results in a very significant problem size reduction for most large circuits. This reduction bears some similarity to the derivation of the pressure-Poisson equation in fluid mechanics [4], and the use of the range-space method in constrained optimization problems [11] . In the case of circuit equations, this reduction does not remove all the constraint equations of the system, so the set of equations is still a set of DAEs, but with index zero.

One issue of interest is that now the variables that are part of the solution vector, $\mathbf{x}$, are no longer the same as the variables for which we need time derivatives. In this particular example, we need the time derivative of the capacitor charge, q, but we do not need time derivatives for V_1, V_2 or I_{Vsrc}. In this document, variables that are part of $\mathbf{x}$ will be referred to as solution variables, while variables that are not part of $\mathbf{x}$ but are needed by the time integration algorithm will be referred to as “state variables”.

This choice of naming convention is consistent with that of the data structures in Spice3f5, but it has the potential to cause confusion. The term, state variable, is often used as the equivalent of the term, solution variable, in much of the circuit simulation literature [9, 7], and other contexts. Additionally, there exists a mathematical formulation of the circuit equations known as the “state variable formulation”, in which the circuit equations are an implicit ODE system. Chua and Lin describe this formulation in detail [6]. This formulation is not commonly used in modern circuit simulators, as it has been proven that some circuits cannot be represented in this form.

The inexact Jacobian used with the condensed form

The condensed form described in the previous section has been implemented with one type of approximation made in the calculation of the Jacobian matrix. There is nothing about the condensed form that requires the use of this approximation, but it facilitates the implementation. This approximation is made in Spice3f5 and **Xyce** for the Jacobian matrix contributions associated with a nonlinear capacitor. The two-terminal nonlinear capacitor contributes the following stencil to the linear system:

$$\begin{bmatrix} & \vdots & & \vdots & \\ \dots & \delta I_C/\delta V_1 & \dots & \delta I_C/\delta V_2 & \dots \\ & \vdots & & \vdots & \\ \dots & -\delta I_C/\delta V_1 & \dots & -\delta I_C/\delta V_2 & \dots \\ & \vdots & & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ \Delta V_1 \\ \vdots \\ \Delta V_2 \\ \vdots \end{bmatrix} = - \begin{bmatrix} \vdots \\ I_C \\ \vdots \\ -I_C \\ \vdots \end{bmatrix} \tag{86}$$

Generally, $\delta I_C/\delta V_1 = -\delta I_C/\delta V_2$, so all the terms in the Jacobian stencil are of the same magnitude. Recall that the capacitor current is given by:

$$I_C = \frac{dq}{dt} \tag{87}$$

For the linear capacitor, this can be simplified to:

$$I_C = \frac{dV_{12}}{dt} \tag{88}$$

Time derivatives are approximated, in general using a BDF:

$$\frac{dV^{i+1}}{dt} = \alpha V^{i+1} + \beta V^i + \gamma V^{i-1} + \dots \tag{89}$$

For the purposes of calculating the Jacobian terms, only the leading term of the BDF is needed, as the partial derivatives used in the matrix are all in terms if new $(i+1)$ variables. So, for the linear capacitor, the form of the Jacobian term is:

$$\frac{\delta I}{\delta V} = \alpha C \tag{90}$$

For the nonlinear capacitor, C is dependent upon the capacitor voltage at $i+1$, so for that case, equation (90) is not correct. However, in Spice3f5 and **Xyce** it is often (but not always) used anyway. For nonlinear problems it is often not necessary that the Jacobian be perfect, and that appears to be the case for circuit problems. However, this may turn out to be inadequate for optimization studies in the future [11] . It is not necessary to use this approximation in the compact state variable formulation, but implementation is easier, as it is not necessary to calculate the partial derivative of C with respect to V. Also, all capacitors types, both linear and nonlinear, are now handled in the same way.

Reduction in Jacobian size resulting form the condensed form

The impact of using the condensed formulation is significant enough that real life circuit codes (such as commercial implementations of SPICE [1], and non-SPICE circuit codes

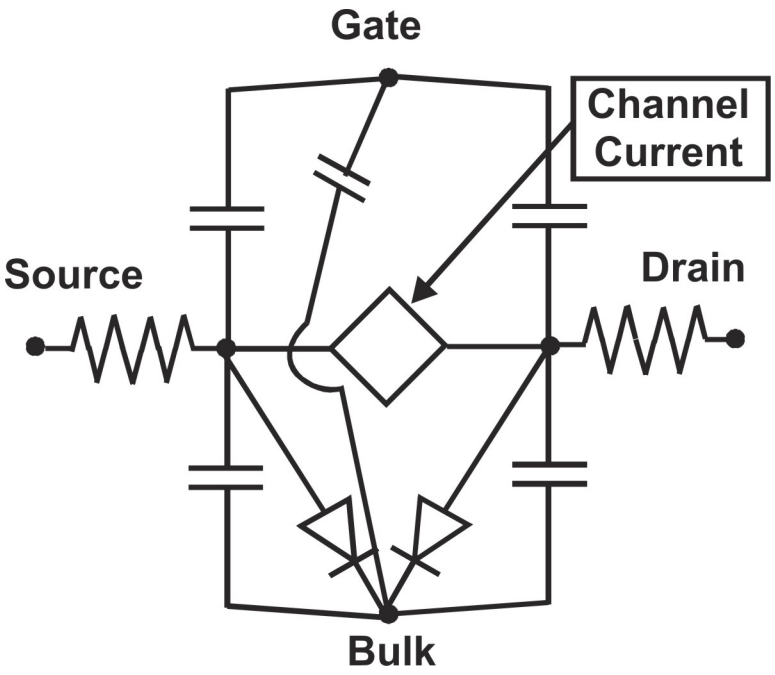

Figure 13. MOSFET model equivalent circuit.

such as SABER [2]) always use it. The Jacobian matrix size can often be reduced by an order of magnitude, especially for highly interconnected digital circuits. For example, most MOSFET models are based on the equivalent circuit shown in Figure 13, which includes five internal nonlinear capacitors.

A large circuit recently simulated in **Xyce** contained about 70,000 MOSFETs, and (comparatively speaking) not much else. If the condensed formulation had not been used, the number of solution variables would have been over 350,000, requiring a 350,000 x 350,000 sparse Jacobian matrix. However, with state variables, the total number of equations was on the order of 25,000, requiring (obviously) a much smaller 25,000 x 25,000 sparse Jacobian matrix. Unlike charge variables, which are not shared between devices, nodal voltage variables can be shared by many devices. As a result, if charge variables are eliminated, the number of solution variables can often be surprisingly small, possibly much smaller than the total number of devices in the circuit.

Additional Notes

As in the case of the nonlinear solver, **Xyce** also has many different options for time integration. Time integration algorithms include backward Euler, BDF2, and Trapezoid rule. Additionally, the time integration package in **Xyce** employs a robust discontinuity capturing scheme. A complete list of current features, as well as instructions, is contained in the **Xyce** User's Guide [8].

It should be noted that currently, the only option in **Xyce** is the condensed DAE form. The option for using the non-condensed traditional formulation, presented in the previous section, is not currently available (**Xyce** Release 2.0). As of this writing, the **Xyce** time integrator is in the process of being redesigned, so other formulations may be available in the future.

References

[1] Hspice user's manual. Technical report, Meta-Software, Inc., Campbell, California, 1996.

[2] Saber mixed signal simulator user's manual. Technical report, Analogy Software, Beaverton, Oregon, 2001.

[3] *NOX and LOCA:Object-Oriented Nonlinear Solver and Continuation Packages*. `http://software.sandia.gov/nox/`, 2004.

[4] Uri M. Ascher and Linda R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, 1988.

[5] K. E. Brenen S. L. Campbell and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Prentice-Hall, North-Holland, New York, 1989.

[6] Leon O. Chua and Pen-Min Lin. *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

[7] Daniel Foty. *MOSFET Modeling with SPICE. Principles and Practice.* Prentice-Hall, Upper Saddle River, New Jersey, 1997.

[8] Scott A. Hutchinson, Eric R. Keiter, Robert J. Hoekstra, Lon J. Waters, Thomas V. Russo, Eric L. Rankin, Roger P. Pawlowski, and Steven D. Wix. Xyce parallel electronic simulator: User's guide, version 2.0. Technical Report SAND2003-xxxx, Sandia National Laboratories, Albuquerque, NM, December 2003.

[9] William Liu. *MOSFET Models for SPICE Simulation including BSIM3v3 and BSIM4*. John Wiley and Sons, New York, 2001.

[10] Laurence Nagel and Ronald Rohrer. Computer analysis of nonlinear circuits, excluding radiation (cancer). *IEEE Journal of Solid-State Circuits*, sc-6(4):166–182, 1971.

[11] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999.

[12] C. W. Ho A. E. Ruehli and P. A. Brennan. The modified nodal approach to network analysis. *IEEE Trans. Circuits Systems*, 22:505–509, 1988.

[13] Gilbert Strang. A framework for equilibrium equations. *SIAM Review*, 30(2):283–297, June 1988.

[14] Jiri Vlach and Kishore Singal. *Computer Methods for Circuit Analysis and Design*. Chapman and Hall, New York, 1994.

Index

DISTRIBUTION:

1 Steven P. Castillo
Klipsch School of Electrical and Computer Engineering
New Mexico State University
Box 3-o
Las Cruces, NM 88003

1 Kwong T. Ng
Klipsch School of Electrical and Computer Engineering
New Mexico State University
Box 3-o
Las Cruces, NM 88003

1 Nick Hitchon
Electrical and Computer Engineering
University of Wisconsin
1415 Engineering Drive
Madison, WI 53706

1 Mark Kushner
Department of Electrical and Computer Engineering
University of Illinois
1406 W. Green Street
Urbana, IL 61801

1 Ron Kielkowski
RCG Research, Inc
8605 Allisonville Rd, Suite 370
Indianapolis, In 46250

1 Mike Davis
Software Federation, Inc.
211 Highview Drive
Boulder, Co 80304

1 Wendland Beezhold
Idaho Accelerator Center
1500 Alvin Ricken Drive
Pocatello, Idaho 83201

1 Kartikeya Mayaram
Department of Electrical and Computer Engineering
Oregon State University
Corvallis, OR 97331-3211

1 Linda Petzold
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106-5070

1 Jaijeet Roychowdhury
4-174 EE/CSci Building
200 Union Street S.E.
University of Minnesota
Minneapolis, MN 55455

1 C.-J. Richard Shi
VLSI and Electronic Design Automation
210 EE/CSE Bldg.
Box 352500
University of Washington
Seattle, WA 98195

1 Homer F. Walker
WPI Mathematical Sciences
100 Institute Road
Worcester, MA 01609

1 Dan Yergeau
CISX 334
Via Ortega
Stanford, CA 94305-4075

1 Masha Sosonkina
319 Heller Hall
10 University Dr.
Duluth, MN 55812

1 Misha Elena Kilmer
113 Bromfield-Pearson Bldg.
Tufts University
Medford, MA 02155

1 Tim Davis
P.O. Box 116120
University of Florida
Gainesville, FL 32611-6120

1 Achim Basermann
C&C Research Laboratories,
NEC Europe Ltd.
Rathausallee 10
D-53757 Sankt Augustin
Germany

1 Philip A. Wilsey
Clifton Labs
7450 Montgomery Road
Suite 300
Cincinnati, Ohio 45236

1 Dale E. Martin
Clifton Labs
7450 Montgomery Road
Suite 300
Cincinnati, Ohio 45236

1 Richard L. Keiter
Department of Chemistry
Eastern Illinois University
600 Lincoln Avenue
Charleston, IL 61920

1 Ellen A. Keiter
Department of Chemistry
Eastern Illinois University
600 Lincoln Avenue
Charleston, IL 61920

1 Lise Keiter-Brotzman
10 College Circle
Staunton, VA 24401

1 Al Lehnen
Mathematics Department
Madison Area Technical College
3550 Anderson Street
Madison, WI 53704

1 Lon Waters
CoMet Solutions, Inc.
11811 Menaul Blvd NE
Suite No. 1
Albuquerque, NM 87112

1 Tim Davis
P.O. Box 116120
University of Florida
Gainesville, FL 32611-6120

1 MS 0321
Bill Camp, 09200

1 MS 0318
Paul Yarrington, 09230

1 MS 1071
Mike Knoll, 01730

1 MS 0310
Robert Leland, 09220

1 MS 0316
Sudip Dosanjh, 09233

1 MS 0525
Paul V. Plunkett, 01734

1 MS 0835
J. Michael McGlaun, 09140

1 MS 0828
Anthony A. Giunta, 09133

1 MS 0139
Stephen E. Lott, 09905

1 MS 0310
Mark D. Rintoul, 09212

1 MS 1110
David Womble, 09214

1 MS 1111
Bruce Hendrickson, 09215

1 MS 0819
Edward Boucheron, 09231

1 MS 0819
Allen C. Robinson, 09231

1 MS 0316
John Aidun, 09235

1 MS 0316
Scott A. Hutchinson, 09233

10 MS 0316
Eric R. Keiter, 09233

1 MS 0316
Deborah Fixel, 09233

1 MS 0316
Robert J. Hoekstra, 09233

1 MS 0316
Brett Bader, 09233

1 MS 0316
Joseph P. Castro, 09233

1 MS 0316
David R. Gardner, 09233

1 MS 0316
Gary Hennigan, 09233

1 MS 0316
Eric Phipps, 09233

1 MS 0316
Eric L. Rankin, 09233

1 MS 0316
Roger Pawlowski, 09233

1 MS 0316
Richard Schiek, 09233

1 MS 1111
John N. Shadid, 09233

1 MS 1111
Andrew Salinger, 09233

1 MS 0316
Paul Lin, 09233

1 MS 0316
Siriphone C. Kuthakun, 09233

1 MS 0807
David N. Shirley, 9328

1 MS 0807
Philip M. Campbell, 9328

1 MS 0847
Scott Mitchell, 09211

1 MS 0847
Mike Eldred, 09211

1 MS 0847
Bart van Bloemen Waanders, 09211

1 MS 0847
Roscoe A. Bartlett, 09211

1 MS 0196
Elebeoba May, 09212

1 MS 1110
Todd Coffey, 09214

1 MS 1110
David Day, 09214

1 MS 1110
Mike Heroux, 09214

1 MS 1110
Pavel B. Bochev, 09214

1 MS 1110
Richard B. Lehoucq, 09214

1 MS 1110
Raymond S. Tuminaro, 09214

1 MS 1110
James Willenbring, 09214

1 MS 1111
Karen Devine, 09215

1 MS 1109
Robert Benner, 09224

1 MS 0316
Harry Hjalmarson, 09235

1 MS 0525
Steven D. Wix, 01734

1 MS 0525
Thomas V. Russo, 01734

1 MS 0525
Regina Schells, 01734

1 MS 0525
Carolyn Bogdan, 01734

1 MS 0525
Mike Deveney, 01734

1 MS 0525
Raymond B. Heath, 01734

1 MS 0525
Ronald Sikorksi, 01734

1 MS 0525
Albert Nunez, 01734

1 MS 1081
Paul E. Dodd, 01762

1 MS 0660
Roger F. Billau, 09519

1 MS 0874
Robert Brocato, 01751

1 MS 1081
Charles E. Hembree, 01739

1 MS 0889
Neil R. Sorenson, 01832

1 MS 0311
Greg Lyons, 02616

1 MS 0311
Martin Stevenson, 02616

1 MS 0328
Fred Anderson, 02612

1 MS 0537
Perry Molley, 02331

1 MS 0537
Siviengxay Limary, 02331

1 MS 0537
John Dye, 02331

1 MS 0537
Barbara Wampler, 02331

1 MS 0537
Doug Weiss, 02333

1 MS 0537
Scott Holswade, 02333

1 MS 0537
George R. Laguna, 02333

1 MS 0481
Joel Brown, 02132

1 MS 0405
Todd R. Jones, 12333

1 MS 0405
Thomas D. Brown, 12333

1 MS 0405
Donald C. Evans, 12333

1 MS 0405
Matthew T. Kerschen, 12333

1 MS 9101
Rex Eastin, 08232

1 MS 9101
Seung Choi, 08235

1 MS 9409
William P. Ballard, 08730

1 MS 9202
Kathryn R. Hughes, 08205

1 MS 9202
Rene L. Bierbaum, 08205

1 MS 9202
Kenneth D. Marx, 08205

1 MS 9202
Stephen L. Brandon, 08205

1 MS 9202
Jason Dimkoff, 08205

1 MS 9202
Brian E. Owens, 08205

1 MS 9401
Donna J. O'Connell, 08751

1 MS 9217
Stephen W. Thomas, 08950

1 MS 9217
Tamara G. Kolda, 08950

1 MS 9217
Kevin R. Long, 08950

1 MS 9915
Mitchel W. Sukalski, 08961

1 MS 1153
Larry D. Bacon, 15333

1 MS 1179
Leonard Lorence, 15341

1 MS 1179
David E. Beutler, 15341

1 MS 1179
Brian Franke, 15341

1 MS 0835
Randy Lorber, 09141

1 MS 1152
Mark L. Kiefer, 01642

1 MS 1137
Greg D. Valdez, 06224

1 MS 1137
Mark A. Gonzales, 06224

1 MS 1138
Rebecca Arnold, 06223

1 MS 1138
Charles Michael Williamson, 06223

1 MS 1138
Harvey C. Ogden, 06223

1 MS 9018
Central Technical Files, 8945-1

2 MS 0899
Technical Library, 9616

1 MS 0612
Review & Approval Desk, for DOE/OSTI, 9612