Streamlined Circuit Device Model Development with fREEDA® and ADOL-C*

Frank P. Hart[1], Nikhil Kriplani[1], Sonali R. Luniya[1],
Carlos E. Christoffersen[2], and Michael B. Steer[1]

[1] Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695-7914. E-mail addresses for the NC State authors are, in order: fphart@eos.ncsu.edu, nmkripla@unity.ncsu.edu, srluniya@unity.ncsu.edu, and m.b.steer@ieee.org.

[2] Department of Electrical Engineering, Lakehead University, 955 Oliver Road, Thunder Bay, Ontario P7B 5E1, Canada. E-mail address: c.christoffersen@ieee.org.

Summary. Time-marching simulation of electronic circuits using the U.C. Berkeley program Spice and variants has been a standard practice for electronics engineers since the mid-1970s. Unfortunately, the development cycle of Spice models may be lengthy because device model equations and their derivatives must be coded manually. Also, many files in the source tree must be modified to define a new model. *f*REEDA®, an object-oriented circuit simulator under development at several universities, overcomes many limitations of the conventional electronic model development paradigm. A key to this implementation is the ADOL-C package, which is used to automatically evaluate the derivatives of the device model equations. As a result models are more compact and the development time is shorter. The development history of selected Spice models and their *f*REEDA® counterparts are presented to illustrate the advantages of this approach. Further information on *f*REEDA® can be found at http://www.freeda.org.

Key words: circuit simulation, semiconductor device modelling, electronic device modelling

1 Introduction

Computer-aided simulation of electronic circuits has been common since the mid-1970s when the U.C. Berkeley program Spice [1] was made available to

*This material is based upon work supported in part by the Space and Naval Warfare Systems Center San Diego under grant number N66001-01-1-8921 as part of the DARPA NeoCAD Program and in part by the U.S. Army Research Laboratory and the U.S. Army Research Office on Multifunctional Adapative Radio Radar and Sensors (MARRS) under grant number DAAD19-01-1-0496.

an electronics industry that was increasingly engaged in the development and manufacture of integrated circuits based on semiconductor devices. In prior decades, most electronics systems were based on collections of discrete semiconductor devices or vacuum tubes which were connected by discrete wires or by printed circuit board wiring. These systems were amenable to relatively low-cost prototype production and laboratory observation of every interconnection point using measuring equipment familiar to electrical engineers. By the mid 1970s, however, integrated circuits with hundreds or thousands of interconnection points – most not observable in a laboratory setting – had rendered the existing prototyping paradigm obsolete. Since then, pre-production validation of integrated circuit designs has relied heavily upon Spice simulation. Spice was not the first simulator produced at U.C. Berkeley, but it did benefit from research done on earlier generations of simulator engines during the 1960s. Over the years, Spice has been ported successfully to generations of less expensive computers. Today it is both economical and common to simulate even discrete circuits implemented on printed circuit boards prior to building prototype models.

Time-marching simulation using state variable-based models was also initially developed in the 1960s [2]. Such simulators formed and solved systems of differential equations. However, these programs fell out of favor because Spice's modelling philosophy led to purely algebraic systems of equations that were inherently more sparse than the state variable approaches. Interest in this approach was renewed when researchers in the discipline of microwave engineering became interested in combining a device's interactions with electromagnetic fields [3]. Spice analyses are limited to voltage and currents only, and so a new simulator environment based on state variable analysis was created to permit this form of analysis. This initial effort has evolved into *f*REEDA® [4]. Presently, *f*REEDA®is the only *netlist-driven* circuit simulator available to the public which uses Automatic Differentiation (AD). One prior effort including AD is disclosed in [5], but this simulator was not netlist-driven. One other effort [6] reported significant results in modelling Metal Oxide Semiconductor transistors with several (AD) tools, but the simulator environment was not made publicly available.

2 Background

2.1 Constitutive Equations and Network Equations

Computer-aided circuit analysis in its most basic form is comprised of two kinds of equations [7, 8], *constitutive* and *network* equations. Constitutive equations usually express voltage as a function of current (in units of Amperes) or vice versa for a *particular* element. Figure 1(a) shows a generic two-terminal element where the voltage across the element is defined with respect to positive and negative terminals, and current is defined positively as flowing from

the positive to the negative terminals. The current has a vector quality in the sense that the arrow in Fig. 1(a) may be reversed and the magnitude negated. Figure 1(b) shows a linear resistor, a simple impedance elements described by Ohm's law, $v(t) = Ri(t)$ [9] or the admittance form, $i(t) = Gv(t)$, where $G = 1/R$. *Network* equations govern the interconnection of elements, and there are two forms based on Kirchoff's Current (KCL) and Voltage (KVL) Laws. Owing to the simplicity of matrix formulation [10, 11] for circuit simulation, the KCL equation form is preferred. The KCL equation form of the matrix is called a Modified Nodal Admittance Matrix (MNAM) [12] because most entries in the matrix have the physical dimension of an admittance (a ratio of current to voltage), but provisions are made through a form of domain decomposition to permit the entry of circuit elements with different physical units. Figure 1(c) shows a network of three generic two-terminal devices. Notice that one node (or terminal) in the circuit is always designated as the reference node so that the MNAM formulation will be non-singular.

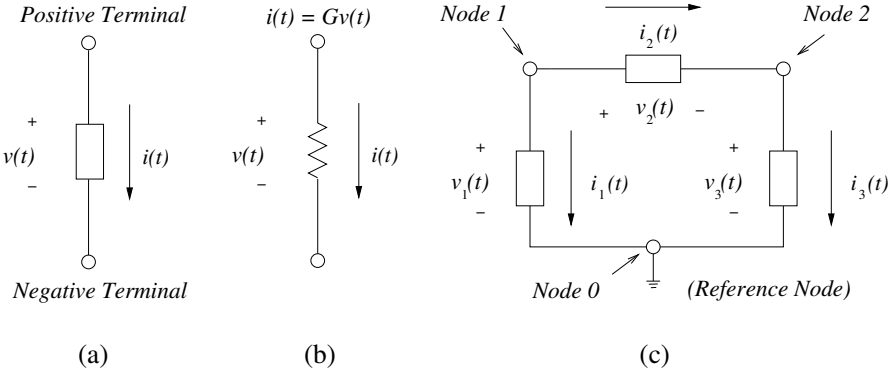


Fig. 1. (**a**) Element constitutive equations; (**b**) Resistor; (**c**) Network equations.

2.2 Forms of Circuit Simulation

The two best-known and longest established forms of circuit simulation are known as *transient* analysis and AC (*i.e.* "alternating current") analysis. Both forms were available in the first release of Spice. Transient analysis is time-domain simulation of the circuit using quadrature integration. Linear and nonlinear differential circuit equations are discretized using either the backward Euler or trapezoidal interpolating functions, and the integration from one time step to the next is governed by an iterative procedure using either Newton's method or the minimization of some error function. AC analysis is frequency-domain simulation of the circuit and is supported only for linear circuit elements in ƒREEDA®, so it will not be discussed further; the emphasis

here is on transient analysis. Underlying both analysis forms is a form known as *DC* (*i.e.* "direct current") analysis, which is used to establish the initial operating conditions for both transient and AC analysis.

One other popular form of circuit simulation that is not present in Berkeley's Spice but has been implemented in *f*REEDA® [13] and other simulators [14, 15] is called *Harmonic Balance* (HB) analysis. HB is an implementation of Galerkin's methods [16] for finding the steady-state response of a nonlinear network. HB will not be discussed in depth, but its method for formulating the MNAM – which differs drastically from that of Spice – has been applied to *f*REEDA®transient analysis [17], and this will be discussed in some detail in Sect. 3.2.

2.3 Constitutive Equations for Elements of Interest

Most transient simulation models are composed of combinations of simpler element models. Those models most useful to the present discussion will be briefly reviewed. Resistors were mentioned in Sec. 2.1. Inductors and capacitors are linear dynamic elements and are described by differential equations [10]. Resistors, capacitors, and inductors often appear within semiconductors in nonlinear forms and are described by Taylor series expansions in these cases.

Nonlinear elements such as semiconductor diodes and transistors are also described by a set of constitutive equations [18]. The diode is perhaps the simplest nonlinear element. Figure 2(a) shows the diode, a device where the current is an exponential function of voltage. A simplified equation for a discrete diode is $i(t) = I_{\rm s}(\exp(v(t)/V_{\rm T}) - 1)$, where $I_{\rm s}$ is known as the reverse saturation current and $V_{\rm T}$, the thermal voltage ($V_{\rm T} \approx 26\,{\rm mV}$ at 300K), is a threshold voltage beyond which the exponential behavior becomes apparent. Figure 2(b) shows the behavior of a typical 1N4153 diode [19]. Figure 3(a)

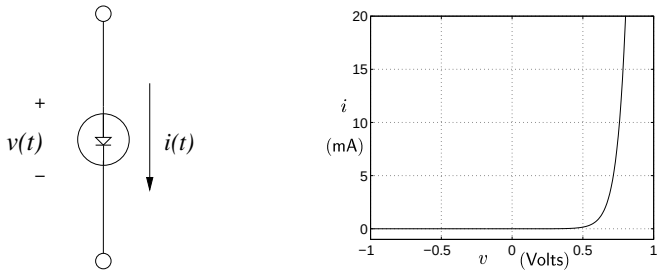


Fig. 2. (**a**) Diode element; (**b**) Current as a function of voltage for a 1N4153 diode.

shows an abstract element form called a 2-port [10], which allows for the arbi-

trary definition of transfer function relationships between two pairs of conductors. For example, $i_1(t) = f(v_1(t), v_2(t), i_2(t))$ or $i_2(t) = f(v_2(t), v_1(t), i_1(t))$. A generalization to an N-port allows for the arbitrary definition of transfer functions from each of N ports to $N-1$ other ports. Port transfer functions are usually provided in matrix form. The constitutive equations for multi-terminal semiconductor devices may be viewed as specific instances of a multi-port.

Finally, circuit simulation often uses two other abstract forms known as ideal voltage and current sources. These are shown in Fig. 3(b–c). They are "ideal" in the sense that their voltages and currents are not related by intrinsic device behavior, but instead are functions of the circuit connections. Specifically, an ideal voltage source has zero internal impedance, so its current is a function of the rest of the circuit. Also, an ideal current source has an infinite internal impedance, so its voltage is a function of the rest of the circuit. These ideal sources are important in equivalent circuit modelling of real devices.

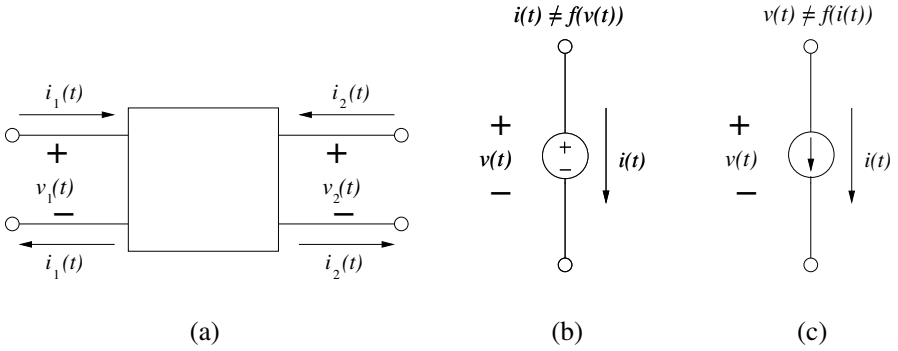


Fig. 3. (**a**) Abstract 2-port element; (**b**) Ideal voltage source; (**c**) Ideal current source.

3 Transient Circuit Simulation Device Modelling

3.1 Spice and Companion Modelling

A 1984 publication reviewing the history of circuit simulation [20] credits Rohrer and a group of graduate students with finding that nonlinear circuit elements could be modelled in the time domain by permitting an equivalent linear circuit of resistors and ideal current sources to have their model values updated not just at time steps within the simulation, but also at different iterates of a Newton iterative loop at a given time step. (Partial disclosure of this technique was given in [21].) This equivalent circuit model was first described as the "Associated Discrete Model" in the literature [7], but more recently it

has been termed simply a "companion model." Companion modelling became the standard method for implementing a nonlinear device model in Spice, and it remains unchanged to this day.

Companion modelling usually features time discretization of the element's equations and incremental linearization of the nonlinear models to permit Newton iteration. The effect of *only* time discretization only can be seen in the companion models for linear capacitors and inductors [7, 22]. Here the companion model for a nonlinear device – a semiconductor diode – will be developed. In the equations that follow, time will be discretized assuming the backward Euler rule and j will indicate the iterate. The companion model begins with the constitutive equation:

$$f(v) \equiv i(t) = I_{\mathrm{s}} \left[\exp\left(\frac{v(t)}{V_{\mathrm{T}}} \right) - 1 \right] . \tag{1}$$

Next, to facilitate Newton iteration, the partial derivative of (1) with respect to v must be obtained:

$$\frac{\partial f}{\partial v} = \frac{I_{\mathrm{s}}}{V_{\mathrm{T}}} \exp\left(\frac{v(t)}{V_{\mathrm{T}}} \right) . \tag{2}$$

Now, (1) and (2) are substituted into (3) defining the Newton iteration, with the result in (4):

$$f(v^{j+1}) = f(v^j + \triangle v) \approx f(v^j) + \frac{\partial f(v^j)}{\partial v^j} \triangle v \Rightarrow \tag{3}$$

$$i^{j+1} = I_{\mathrm{s}} \left[\exp\left(\frac{v^j}{V_{\mathrm{T}}} \right) - 1 \right] + \frac{I_{\mathrm{s}}}{V_{\mathrm{T}}} \exp\left(\frac{v^j}{V_{\mathrm{T}}} \right) (v^{j+1} - v^j) . \tag{4}$$

From the form of (4), it can be seen that the result is an equivalent circuit consisting of an ideal current source (with value set during the previous iteration) and a resistor. These terms are identified in (5) and (6). Note that time is not explicitly discretized in this model, but time discretization is implied in the iterated voltage values, which are functions of time.

$$I_{\mathrm{eq}}^j = I_{\mathrm{s}} \left[\exp\left(\frac{v^j}{V_{\mathrm{T}}} \right) - 1 \right] - \frac{I_{\mathrm{s}} v^j}{V_{\mathrm{T}}} \exp\left(\frac{v^j}{V_{\mathrm{T}}} \right) \tag{5}$$

$$g_{\mathrm{eq}}^j = \frac{I_{\mathrm{s}}}{V_{\mathrm{T}}} \exp\left(\frac{v^j}{V_{\mathrm{T}}} \right) . \tag{6}$$

Consider now a simple circuit consisting of an ideal current source, a resistor, and a diode as shown in Fig. 4(a). In this circuit, there is only one node other than the reference node, and thus the Newton iteration is on only one equation. It is still illustrative, but bear in mind that circuits with more than one node will require Newton iteration on vectors, and this will require Jacobian matrices. For this simple circuit, the analysis task is to determine

the currents through the resistor and diode at all times. Note that the resistor current is dependent on the diode voltage, so that the solution depends wholly upon the diode. The Spice transient analysis method substitutes the companion model for the diode, transforming the circuit into the collection of resistors and current sources shown in Fig. 4(b). Equations (7–8) show the results of the companion model substitution. Applying KCL at the top node (or terminal) of Fig. 4,

$$I_{\rm src} = Gv^{j+1} + i^{j+1} \Rightarrow i^{j+1} = I_{\rm src} - Gv^{j+1} = I^j_{\rm eq} + g^j_{\rm eq} v^{j+1} \tag{7}$$

$$\Rightarrow v^{j+1} = \frac{I_{\rm src} - I^j_{\rm eq}}{G + g^j_{\rm eq}} \, . \tag{8}$$

The Spice transient analysis routine will perform Newton iteration at every time step, updating the MNAM with new g^j_{eq} values and the vector of sources with new I^j_{eq} values at every iteration until convergence. At each iterate, the time step is reduced so that the values of v^{j+1} approach v^j.

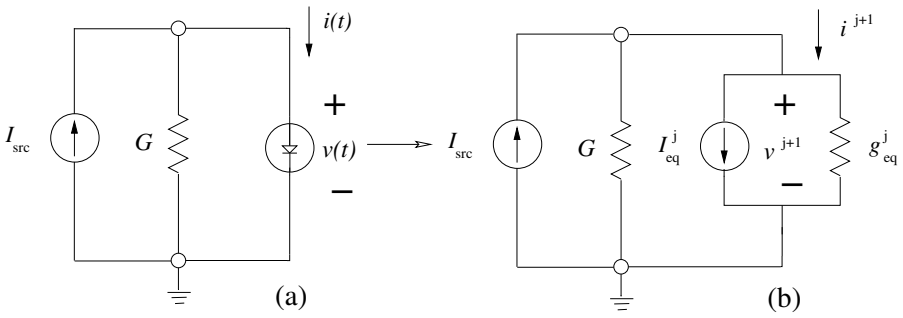


Fig. 4. (**a**) Simple circuit containing a diode; (**b**) Equivalent circuit containing companion model.

The obvious advantage of companion modelling is now apparent. Through companion model substitution, nonlinear behavior is reduced is reduced to a simple linear form. In addition, Spice transient analysis is reduced to a simple extension of Spice DC analysis with Newton iteration. However, the algebraic model form must be obtained by mathematical analysis and inserted into the model code, and this form is not always easily obtained. Moreover, an unfortunate consequence of updating model values as a simulation progresses is that the MNAM contents change at every time step and at every Newton iterate within a time step, and so the computationally expensive LU factorization performed on the MNAM gets no reuse. Philosophically, the companion model is a mathematical abstraction that risks departing from the underlying physics of the device in order to facilitate a simulation method consisting of equivalent circuit models with very few elements.

3.2 fREEDA®and State Variable Based Modelling Using ADOL-C

In *f*REEDA®, device element models are faithful replicas of the physical equations describing the device. Consider the currents and voltages at the ports of a nonlinear device to be expressed as functions of independent parameters called state variables ($\mathbf{x}()$), *i.e.*

$$\mathbf{v}_{\rm p}(t) = \mathbf{v}[\mathbf{x}(t), \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}, \cdots, \frac{\mathrm{d}^m\mathbf{x}}{\mathrm{d}t^m}, \mathbf{x}_{\rm D}(t)] \tag{9}$$

$$\mathbf{i}_{\rm p}(t) = \mathbf{i}[\mathbf{x}(t), \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}, \cdots, \frac{\mathrm{d}^m\mathbf{x}}{\mathrm{d}t^m}, \mathbf{x}_{\rm D}(t)] \ . \tag{10}$$

Here, $\mathbf{x}_{\rm D}(t)$ is a time-delayed version of $\mathbf{x}(t)$. To avoid charge conservation problems in transient analysis [23], (9–10) must be reformulated in stages as follows:

$$\text{stage } 1: \begin{cases} \mathbf{f}_1(\mathbf{x}(t), \mathbf{x}_{\rm D}(t)) \\ \mathbf{g}_1(\mathbf{x}(t), \mathbf{x}_{\rm D}(t)) \end{cases} \tag{11}$$

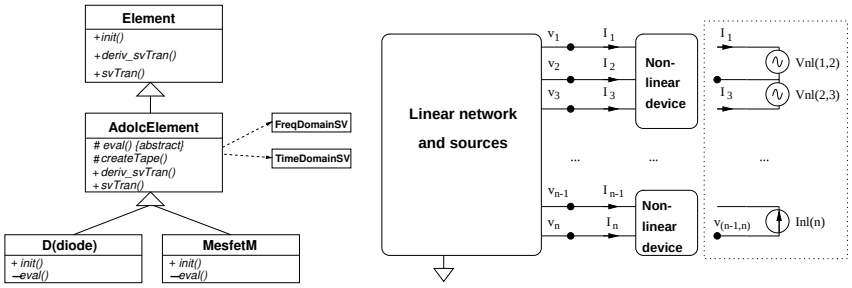
$$\text{stage } 2: \begin{cases} \mathbf{f}_2(\mathbf{f}_1(t), \mathrm{d}\mathbf{g}_1/\mathrm{d}t) \\ \mathbf{g}_2(\mathbf{f}_1(t), \mathrm{d}\mathbf{g}_1/\mathrm{d}t) \end{cases} \tag{12}$$

$$\vdots$$

$$\text{stage } n-1: \begin{cases} \mathbf{f}_{n-1}(\mathbf{f}_{n-2}(t), \mathrm{d}\mathbf{g}_{n-2}/\mathrm{d}t) \\ \mathbf{g}_{n-1}(\mathbf{f}_{n-2}(t), \mathrm{d}\mathbf{g}_{n-2}/\mathrm{d}t) \end{cases} \tag{13}$$

$$\text{stage } n: \begin{cases} \mathbf{v}(\mathbf{f}_{n-1}(t), \mathrm{d}\mathbf{g}_{n-1}/\mathrm{d}t) \\ \mathbf{i}(\mathbf{f}_{n-1}(t), \mathrm{d}\mathbf{g}_{n-1}/\mathrm{d}t) \end{cases} \ . \tag{14}$$

An example and advantages of this formulation will be given in Sect. 4.3. All arguments in (11–14) become ADOL-C active variables [24] so that the derivatives of functions $\mathbf{f}_1$, $\mathbf{g}_1$, $\mathbf{f}_2$, *etc.* can be obtained automatically and used in the model code. Derivatives are calculated in the forward mode in *f*REEDA®. Comparison of forward and reverse modes is a matter for future research. Through the use of object-oriented programming techniques, all device models are derived C++ classes that inherit the characteristics of a C++ base class [4]. For nonlinear devices in *f*REEDA®, "AdolcElement" is the base class, and other nonlinear devices are derived classes as illustrated in the Unified Modelling Language (UML) diagram in Fig. 5(a). Model developers describe the number of terminals and state variables required for the element in the *init()* function of the derived class and implement the nonlinear equations unique to the derived class in the *eval()* function for the derived class. If more than one stage is necessary, they are implemented in functions called *eval2()* and *eval3()*. The AdolcElement class contains the interface to ADOL-C for initializing and manipulating the ADOL-C 'tapes' of active variables. AdolcElement also encapsulates and hides details of working with ADOL-C. Calls to the *eval()* routine for the derived class are bracketed between ADOL-C *trace* statements in the AdolcElement code. The calculation of total derivatives and time delayed variables is performed by the TimeDomainSV or the



(a) (b)

Fig. 5. (**a**) UML diagram; (**b**) Transient simulation circuit partitioning.

FreqDomainSV classes, depending on the type of analysis being performed. Thus, the same code can be used in any circuit analysis. This is made possible by the fact that time is discretized in the ƒREEDA® transient analysis code and not in the device model code. ADOL-C is used to calculate the derivatives of the functions at each stage and thus obtain the Jacobian of the currents and voltages with respect to the state variables (as shown in [4]). The procedure to obtain the Jacobian is embedded in AdolcElement so that model developers need not code that in their derived classes. Absent ADOL-C, a state-variable approach would require manual coding of the Jacobian matrices. Thus the importance of the ADOL-C package to facilitating this state variable approach in ƒREEDA® cannot be understated. The results of evaluating (9–10) for each port are collected by the ƒREEDA® state variable transient analysis routine into a vector of equations describing the nonlinear portion of the circuit:

$$\mathbf{v}_{\mathrm{NL}}(t) = \mathbf{v}[\mathbf{x}(t), \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}, \cdots, \frac{\mathrm{d}^m\mathbf{x}}{\mathrm{d}t^m}, \mathbf{x}_{\mathrm{D}}(t)] \tag{15}$$

$$\mathbf{i}_{\mathrm{NL}}(t) = \mathbf{i}[\mathbf{x}(t), \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}, \cdots, \frac{\mathrm{d}^m\mathbf{x}}{\mathrm{d}t^m}, \mathbf{x}_{\mathrm{D}}(t)]\ . \tag{16}$$

The formulation of the system of equations for ƒREEDA®'s transient analysis may now be described. Figure 5(b) shows the partitioning of the complete circuit into linear and nonlinear portions, with the port abstractions for each nonlinear device indicated. At the boundary between the linear and nonlinear portions, the voltages must be equal. Thus two voltage vectors, one a function of the linear circuit behavior and the other a function of the nonlinear circuit behavior, must be equal and can be used to form an error function. Let the linear portion of the circuit in Fig. 5(b) be described by two MNAMs, $\mathbf{G}$ and $\mathbf{C}$, where $\mathbf{G}$ describes the linear static elements (such as resistors) and $\mathbf{C}$ describes the linear dynamic elements (such as linear capacitors and inductors). Also, let $\mathbf{u}$ be the vector of unknown voltages and currents and $\mathbf{s}$ be a vector of sources. Then,

$$\mathbf{G}\mathbf{u}(t) + \mathbf{C}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{s}(t) \tag{17}$$

$$\mathbf{s}(t) = \mathbf{s}_{\mathrm{f}}(t) + \mathbf{s}_{\mathrm{v}}(t)\ . \tag{18}$$

In (18), the source vector $\mathbf{s}$ is comprised of $\mathbf{s}_{\mathrm{f}}$, a vector of independent *forcing* sources, and $\mathbf{s}_{\mathrm{v}}$, a vector of currents injected by the nonlinear circuit into the linear circuit. Now, through the use of an incidence matrix, $\mathbf{T}$[3], which specifies connectivity information relating the circuit's node assignments to its state variables, the following relationships hold:

$$\mathbf{v}_{\mathrm{L}}(t) = \mathbf{T}\mathbf{u}(t) \tag{19}$$

$$\mathbf{s}_{\mathrm{v}}(t) = \mathbf{T}^{\mathrm{T}}\mathbf{i}_{\mathrm{NL}}(t)\ . \tag{20}$$

In (19), $\mathbf{v}_{\mathrm{L}}(t)$ is the vector of port voltages from the linear elements at the linear/nonlinear interface boundary. Substituting (20) into (17–18) and simplifying yields the equation for the state of the system:

$$\mathbf{G}\mathbf{u}(t) + \mathbf{C}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{s}_{\mathrm{f}}(t) + \mathbf{T}^{\mathrm{T}}\mathbf{i}_{\mathrm{NL}}(t)\ . \tag{21}$$

Equation (21) is subject to the equation for the error function:

$$\mathbf{f}(t) = \mathbf{v}_{\mathrm{L}}(t) - \mathbf{v}_{\mathrm{NL}}(t) = \mathbf{0} \tag{22}$$

$$\mathbf{f}(t) = \mathbf{T}\mathbf{u}(t) + \mathbf{v}_{\mathrm{NL}}(t) = \mathbf{0}\ . \tag{23}$$

It was noted earlier in this section that time discretization does not occur within the element class definition in *f*REEDA®; instead it occurs within the analysis routine. This has the dual advantages of simplifying the coding of the element classes and also allowing for different time interpolating functions. *f*REEDA®allows a choice of either Backward Euler or Trapezoidal interpolating functions. After discretizing the vector of unknowns $\mathbf{u}(t)$, its derivative $\mathrm{d}\mathbf{u}(t)/\mathrm{d}t$ is defined as

$$\frac{\mathrm{d}\mathbf{u}(t)}{\mathrm{d}t} \Rightarrow \mathbf{u}'_n = a\mathbf{u}_n + \mathbf{b}_{n-1}\ , \tag{24}$$

where the scalar a and vector $\mathbf{b}$ depend upon the choice of interpolating function. Substituting the discretized $\mathbf{u}$ into (21),

$$\mathbf{G}\mathbf{u}_n + \mathbf{C}[a\mathbf{u}_n + \mathbf{b}_{n-1}] = \mathbf{s}_{\mathrm{f},n} + \mathbf{T}^{\mathrm{T}}\mathbf{i}_{\mathrm{NL}}(\mathbf{x}_n)\ , \tag{25}$$

and solving for $\mathbf{u}_n$,

$$\mathbf{u}_n = [\mathbf{G} + a\mathbf{C}]^{-1}[\mathbf{s}_{\mathrm{f},n} - \mathbf{C}\mathbf{b}_{n-1} + \mathbf{T}^{\mathrm{T}}\mathbf{i}_{\mathrm{NL}}(\mathbf{x}_n)]\ . \tag{26}$$

[3]The number of rows of $\mathbf{T}$ is equal to the total number of nonlinear ports and the number of columns is equal to the number of nodes. For each row, a $+1$ entry denotes the $+$ terminal of a port, and a -1 denotes the $-$ terminal. All other entries are 0.

Discretizing the error function defined in (22-23),

$$\mathbf{f}(\mathbf{x}_n) = \mathbf{T}\mathbf{u}_n - \mathbf{v}_{\mathrm{NL}}(\mathbf{x}_n) = \mathbf{0}\ . \tag{27}$$

Now, define the following quantities comprised of $\mathbf{T}\mathbf{u}_n$:

$$\mathbf{s}_{\mathrm{sv},n} = \mathbf{T}[\mathbf{G} + a\mathbf{C}]^{-1}[\mathbf{s}_{\mathrm{f},n} - \mathbf{C}\mathbf{b}_{n-1}] \tag{28}$$

$$\mathbf{M}_{\mathrm{sv}} = \mathbf{T}[\mathbf{G} + a\mathbf{C}]^{-1}\mathbf{T}^{\mathrm{T}}\ . \tag{29}$$

Substituting (28–29) into (27) leads to

$$\mathbf{f}(\mathbf{x}_n) = \mathbf{s}_{\mathrm{sv},n} + \mathbf{M}_{\mathrm{sv}}\mathbf{i}_{\mathrm{NL}} - \mathbf{v}_{\mathrm{NL}}(\mathbf{x}_n) = \mathbf{0}\ . \tag{30}$$

Note that for a fixed step size – usually the case – $\mathbf{M}_{\mathrm{sv}}$ is a constant and the matrix $[\mathbf{G} + a\mathbf{C}]^{-1}$ appearing in (28–29) is LU-factored only once per simulation.

4 Selected Modelling Examples

4.1 Illustrative Comparison of Diode Models

The development of models for the semiconductor diode makes for an illustrative comparison. A full description of aspects of the Spice diode model development is given in [22]. Details of the *f*REEDA® diode model are described in [25]. The Spice companion model for a simplified diode model was described in Sect. 3.1, and it was noted there that the process for developing companion models requires model developers to perform derivatives on the constitutive equations and manually code the derivative equations to facilitate Newton iteration. The *f*REEDA® code for the *init()* and *eval()* functions for the simplified two parameter diode derived class is shown in Listing 1. In this case, the *eval()* function is literally coding the constitutive equations. These two functions take up only 17 lines of code. The complete C++ code and header files for the simplified diode are both only 56 lines long.

```
void DiodeJcn::init() throw(string&) {
  // Set the number of terminals
  setNumTerms(2);
  // Set number of states
  setNumberOfStates(1);
  // create tape
  IntVector var(1,0); // create vector of 1 element set to 0
  createTape(var);    // creates state var x[0]
}

void DiodeJcn::eval(adoublev& x,
           adoublev& vp, adoublev& ip)
{
```

```
  // x[0] == input voltage
  vp[0] = x[0];
  ip[0] = is * ( exp(x[0]/vT) - 1 );
}
```

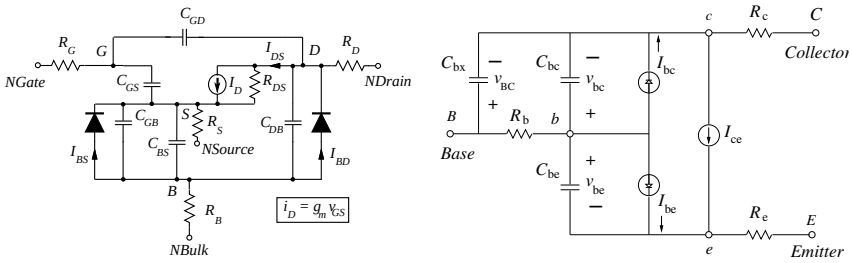
Listing 1. Critical model code for simplified diode.

4.2 The Berkeley Short-channel IGFET Model Version 4 (BSIM4)

The Device Group at the University of California at Berkeley has been at the forefront of specifying semiconductor physics models for field effect transistors in the most advanced semiconductor process technologies. Figure 6(a) shows the model for a contemporary Metal Oxide Semiconductor Field Effect Transistor (MOSFET) device. The first Field Effect Transistor (FET) models were published in 1968 and had only 41 parameters to fully describe any transistor. In the 1980s, as semiconductor process technologies continued to shrink the minimum size of their features, other physical phenomena were observed which necessitated the addition of more parameters to the models, and the BSIM models were created.

In 2000, a fourth major version of the BSIM models called BSIM4 was released. Figure 6(a) shows the model for a contemporary short-channel MOSFET device consisting of four terminals (gate, drain, source, and bulk). The BSIM4 device semiconductor physics model has over 200 parameters, and the current high level of interest in this model makes it a good choice for a case study comparison of a Spice model with a *f*REEDA®model. Such a study was completed by one of the authors, who implemented the BSIM4 model in *f*REEDA®as a Master's Thesis [26] in 2002. Starting with the BSIM4 semiconductor physical model documentation, the model consisted of about 1500 lines of C++ code, fitting in two files and on 25 printed pages, and required 7 months to develop. Much of this time was spent implementing code structures to support particular features of the BSIM models for the first time.

Due to industry involvement with UC-Berkeley in the development of the BSIM4 model, it is not possible to make precise development cycle comparisons with *f*REEDA®, but from archival materials at UC-Berkeley, it can be observed [27] that 21 months passed from the final BSIM3 release until the first BSIM4 release. It is also known that the contemporary BSIM4 models consist of about 20,000 lines of C code spanning 21 source code files [28]. It should be noted that Berkeley's MOSFET models are not the only ones available, and implementing other models has proven to be less labor-intensive. For example, ETH-Switzerland's EPFL-EKV model [29], which has only 44 parameters, was implemented by one student [30] as a semester project for a circuit simulation class at NC State University. At the time, the student had limited knowledge of *f*REEDA®'s internals.



(a) (b)

Fig. 6. Equivalent circuit model schematic diagrams for the (**a**) MOSFET; (**b**) Bipolar Junction Transistor (BJT).

4.3 fREEDA® Universal Modelling Approach

A universal state variable-based modelling approach was proposed in [23]. The central notion of this approach is to choose as state variables the quantities most appropriate to accurately model the physical device as illustrated in Sect. (3.2), and then derive any additional required variables through a set of hierarchically evaluated functions. To illustrate the approach, consider the simplified NPN-type Bipolar Junction (BJT) model of Fig. 6(b). By applying the universal modelling approach, it is possible to create a device model for the BJT that conserves charge [23]. Charge conservation has been a problem with some Spice MOSFET models [31, 32], thus rendering the models suspect to semiconductor physicists. Referring to Fig. 6(b), the voltages across the base collector capacitor $v_{\rm bc}$ and base emitter capacitor $v_{\rm be}$ are chosen as state variables (or independent variables). A charge-conserving model of the bipolar transistor is then described using three ADOL-C tapes. Each tape has two input and two output variables. In tape 1 the quiescent current components $I_{\rm bc}$ and $I_{\rm be}$ are computed as

$$I_{\rm be} = \frac{I_{\rm bf}}{\beta_{\rm F}} + I_{\rm le} \tag{31}$$

$$I_{\rm bc} = \frac{I_{\rm br}}{\beta_{\rm R}} + I_{\rm lc}\ , \tag{32}$$

where $\beta_{\rm F}$ is a current gain parameter, $I_{\rm bf}/\beta_{\rm F}$ and $I_{\rm le}$ are the components of the currents through the base-emitter diode, and $I_{\rm br}/\beta_{\rm F}$ and $I_{\rm lc}$ are components of the the currents through the base-collector diode. The first output vector from tape 1 stores the charge across the base collector and base emitter capacitors which can be evaluated as

$$q_{\rm bc}(v) = \int_0^v c_{\rm bc}(v_{\rm bc}){\rm d}v_{\rm bc} \tag{33}$$

$$q_{\rm be}(v) = \int_0^v c_{\rm be}(v_{\rm be}){\rm d}v_{\rm be} \ , \tag{34}$$

where the integrals are evaluated analytically. The second output vector stores the diode current components and junction voltages. The derivatives of the charge across the capacitors,

$$I_{\rm Cbc} = \frac{{\rm d}q_{\rm bc}}{{\rm d}t} \tag{35}$$

$$I_{\rm Cbe} = \frac{{\rm d}q_{\rm be}}{{\rm d}t} \ , \tag{36}$$

are used as input parameters in tape 2. These derivatives are obtained with a formula that depends of the type of quadrature integration being used. The approximation of the time derivatives are used to calculate the charge across the distributed base collector capacitor $C_{\rm bx}$ in a manner similar to that done in (33-34). Inputs to tape 3 contain the corresponding charge in $C_{\rm bx}$ and the junction voltages. In tape 3 the current across $C_{\rm bx}$ is computed in a manner analogous to (35-36), and the final external voltages and currents are calculated using the intermediate variables generated at the previous tapes.

The ADOL-C tapes are generated once and then used in the main program every time the bipolar transistor model equations or its derivatives with respect to its input parameters need to be evaluated. The procedure to calculate the derivatives of the model equations from the set of tapes is the same for all models [23] and is handled by a base class common to all elements. Therefore the addition of the bipolar transistor model in *f*REEDA® is accomplished by adding a new derived class (one ".cc" file and a header file) with the definition of the three ADOL-C tapes plus other information such as the number of terminals and model parameter names. A hierarchical state-variable approach similar to that shown here can be applied to assure charge conservation in other nonlinear devices, and it is advocated for all nonlinear devices.

5 Conclusion

Through a deft combination of the use of ADOL-C's automatic differentiation capabilities, object-oriented programming techniques to encapsulate and hide much of the ADOL-C interfacing details from the model code, and choosing to discretize time outside of device model code, *f*REEDA® has dramatically eased nonlinear circuit device model development. The guiding philosophy behind *f*REEDA® is to facilitate the implementation of models as close to the physics of devices as possible, and in most cases it is possible to literally code the constitutive equations for the device. *f*REEDA® is freely available under GNU Public License at `http://www.freeda.org`.

References

1. L.W. Nagel and R.A. Rohrer. SPICE2, A computer program to simulate semiconductor circuits. Technical Report ERL-M520, University of California at Berkeley, May 1975.
2. C. Pottle. A "Textbook"computerized state-space network analysis algorithm. *IEEE Transactions on Circuit Theory*, 16(4):566–568, November 1969.
3. F. Nusseibeh, T.W. Nuteson, J. Patwardhan, M.A. Summers, C.E. Christoffersen J. Kreskovsky, and M.B. Steer. Computer-aided engineering environment for spatial power combining systems. In *1997 IEEE MTT-S International Microwave Symposium Digest*, volume 2, pages 1073–1076. IEEE MTT-S, IEEE Press, June 1997.
4. C.E. Christoffersen, U.A. Mughal, and M.B. Steer. Object oriented microwave circuit simulation. *International Journal of RF and Microwave Computer-Aided Engineering*, 10(3):164–182, May 2000.
5. B. Melville, P. Feldmann, and S. Moinian. A C++ based environment for analog circuit simulation. In *IEEE 1992 International Conference on Computer Design*, pages 516–519. IEEE Press, October 1992.
6. W. Klein. Comparisons of different automatic differentiation tools in circuit simulation. In *Computational Differentiation: Techniques, Applications, and Tools*, chapter 26, pages 297–307. Society for Industrial and Applied Mathematics, 1996.
7. L.O. Chua and P.M. Lin. *Computer-Aided Analysis of Electronic Circuits: Algorithms & Computational Techniques*. Prentice-Hall, 1975.
8. J. Ogrodzki. *Circuit Simulation Methods and Algorithms*. CRC Press, 1994.
9. D. Halliday and R. Resnick. *Fundamentals of Physics*. John Wiley and Sons, 2 edition, 1981.
10. C.A. Desoer and E.S. Kuh. *Basic Circuit Theory*. McGraw-Hill, 1969.
11. J. Vlach and K. Singhal. *Computer Methods for Circuit Analysis and Design*. Van Nostrand Reinhold, 1994.
12. C.W. Ho, A.E. Ruehli, and P.A. Brennan. The modified nodal approach to network analysis. *IEEE Transactions on Circuits and Systems*, CAS-22(6):504–509, June 1975.
13. C.E. Christoffersen, M.B. Steer, and M.A. Summers. Harmonic balance analysis for systems with circuit-field iterations. In *1998 IEEE MTT-S International Microwave Symposium Digest*, volume 2, pages 1131–1134. IEEE MTT-S, IEEE Press, June 1998.
14. K.S. Kundert, J.K. White, and A. Sangiovanni-Vincentelli. *Steady-State Methods for Simulating Analog and Microwave Circuits*. Kluwer Academic Publishers, 1990.
15. P.J.C. Rodrigues. *Computer-Aided Analysis of Nonlinear Microwave Circuits*. Artech House, 1998.
16. A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*, volume 37 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 2000.
17. C.E. Christoffersen. *Global Modeling Of Nonlinear Microwave Circuits*. Ph.D Dissertation, North Carolina State University, 2000.
18. B.G. Streetman. *Solid-State Electronic Devices*. Prentice-Hall, 2 edition, 1980.
19. J. Millman. *Microelectronics*. McGraw-Hill, 1979.
20. D.O. Pederson. A historical review of circuit simulation. *IEEE Transactions on Circuits And Systems*, CAS-31(1):103–111, January 1984.

21. W.J. McCalla and W.G Howard. Bias-3 - a program for the nonlinear dc analysis of bipolar transistor circuits. *IEEE Journal of Solid-State Circuits*, SC-6(1):14–19, February 1971.
22. L.T. Pillage, R.A. Rohrer, and C. Visweswariah. *Electronic Circuit System Simulation Methods*. McGraw-Hill, 1995.
23. C.E. Christoffersen, S. Velu, and M.B. Steer. A universal parameterized nonlinear device model formulation for microwave circuit simulation. In *2002 IEEE MTT-S International Microwave Symposium Digest*, volume 3, pages 2189–2192. IEEE MTT-S, IEEE Press, June 2002.
24. A. Griewank, D. Juedes, H. Mitev, J. Utke, O. Vogel, and A. Walther. ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. Technical report, March 1999.
25. C.E. Christoffersen. State variable harmonic balance simulation of a quasi-optical power combining system. Master's Thesis, North Carolina State University, 1998.
26. N.M. Kriplani. Transistor modeling using advanced circuit simulator technology. Master's Thesis, North Carolina State University, 2002.
27. UC-Berkeley Device Group. BSIM Homepage. http://www-device.eecs.berkeley.edu/ bsim3/, 2004.
28. L. Lemaitre, C. McAndrew, and S. Hamm. ADMS – automatic device model synthesizer. In *Proceedings of the IEEE 2002 Custom Integrated Circuits Conference*, volume 1, pages 27–30. IEEE Press, May 2002.
29. M. Bucher, C. Lallement, C. Enz, F. Theodoloz, and F. Krummenacher. The EPFL-EKV MOSFET model equation for simulation. Technical Report Model Version 2.6, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, June 1997.
30. W. Jang. Ekv.doc. See the Mosnekv .cc and .h files in a *f*REEDA® installation, April 2003.
31. M.A. Cirit. The Meyer model revisited: Why is charge not conserved? *IEEE Transactions on Computer-Aided Design*, 8(10):1033–1037, October 1989.
32. P. Yang, B.D. Epler, and P.K. Chaterjee. An investigation of the charge conservation problem for mosfet circuit simulation. *IEEE Journal of Solid-State Circuits*, SC-18(1):128–139, February 1983.

Index