TIME-DOMAIN NETWORK ANALYSIS

Time-domain analysis of nonlinear networks is a complicated process composed of several steps. To avoid inaccuracies and possible confusion, this article covers all such steps, starting with a few definitions. Standard network elements are reviewed in the section entitled "Basic Concepts and Definitions" in a form suitable for network formulations. The section entitled "Kirchhoff's Laws" briefly summarizes Kirchhoff's laws and introduces the concept of cuts, needed later in state variables. The section entitled "Nodal and Loop Equations" repeats nodal and loop equations, a concept taught in every course of network analysis. The two methods are suitable for hand solutions, but are not sufficiently general for computer applications.

Network equations are normally solved by triangular decomposition, a method which may not be known to the reader. A very brief summary is provided in the section entitled "Solutions of Network Equations," along with the Newton–Raphson method, used for solution of nonlinear equations. The section entitled "Graphs" covers basic concepts of graph theory, to the extent needed later for the various methods of collecting the network equations. One such method, state variables, is covered in the section entitled "State Variables." It is still used for theoretical studies but is not suitable for computer applications, and we explain the difficulties. A more general method, the tableau, is explained in the section entitled "Tableau." It keeps as many equations as possible, and it pays the price by leading to very large systems. The method is useful only if a complicated sparse matrix solver is available. The best method to write network equations is the modified nodal. It is covered in the section entitled "Modified Nodal Formulation," to an extent sufficient for understanding. References will help the reader in further studies.

Time-domain solutions are done by methods which replace derivatives by special expressions. The subject is divided into two parts. The section entitled "Simple Integration Methods" explains three simpler methods, where various problems can be easily explained. Although they are simple, they are extensively used in commercial simulators and are quite practical. More advanced integration formulas are covered in the section entitled "Advanced Integration Methods," where we concentrate on the modern backward differentiation formulas.

With these preliminary steps we are in the position to explain time-domain solutions. The subject is divided into two parts. The section entitled "Linear Networks" deals with the integration of linear algebraic differential equations and derives simple formulas which are easy to use. The section entitled "Nonlinear Networks" explains integration methods for nonlinear networks. It points out that nonlinear capacitors and inductors must be described by their charge and flux equations, and it shows how to formulate the Newton–Raphson iteration procedure.

Recent advances in semiconductor technology made it possible to use transistors as switches. They are reliable and fast and opened completely new areas. The section entitled "Switched Networks" introduces the reader to the problem of switched networks and offers a simple solution how to analyze switched networks in time domain.

The bibliography at the end of this article lists publications for additional study.

BASIC CONCEPTS AND DEFINITIONS

Solutions of networks require unified notation for which we need the necessary definitions. Most of them are known to the reader, but some definitions and expressions may somewhat differ. To start we define the concept of ground, node, and electrical element.

Ground usually refers to our earth, but for network analysis it is the chassis or the metal construction into which the electric network is built.

Electrical element is any product functioning in the network. The simplest elements will be defined in this section. The function of most electronic devices, like transistors, can be described for the purposes of network analysis by a collection of these basic elements.

Node is a point where two or more elements are electrically connected together. If the node can be accessed from the outside, then it is called the *terminal.*

With these basic definitions we can turn our attention to the concept of voltage and current.

Voltage is electrical force which is applied across some element and which drives the flow of electrons through the element. The simplest source of voltage is a battery with its + and − terminal. If any electrical element is connected to these two terminals, some amount of electrons will flow through the element and this flow is called the *current.* The definition of current direction was introduced long before the existence of electrons was discovered, and the accepted *positive* direction of current in network analysis is opposite to the flow of electrons: from the more positive point (or from +) to a less positive point (or to −). For general voltages within the network we will use the letter V, and for currents flowing in the network we will use the letter I.

Ideal voltage source is a fundamental element in network analysis, and its symbol is in Fig. 1. Positive direction of cur-

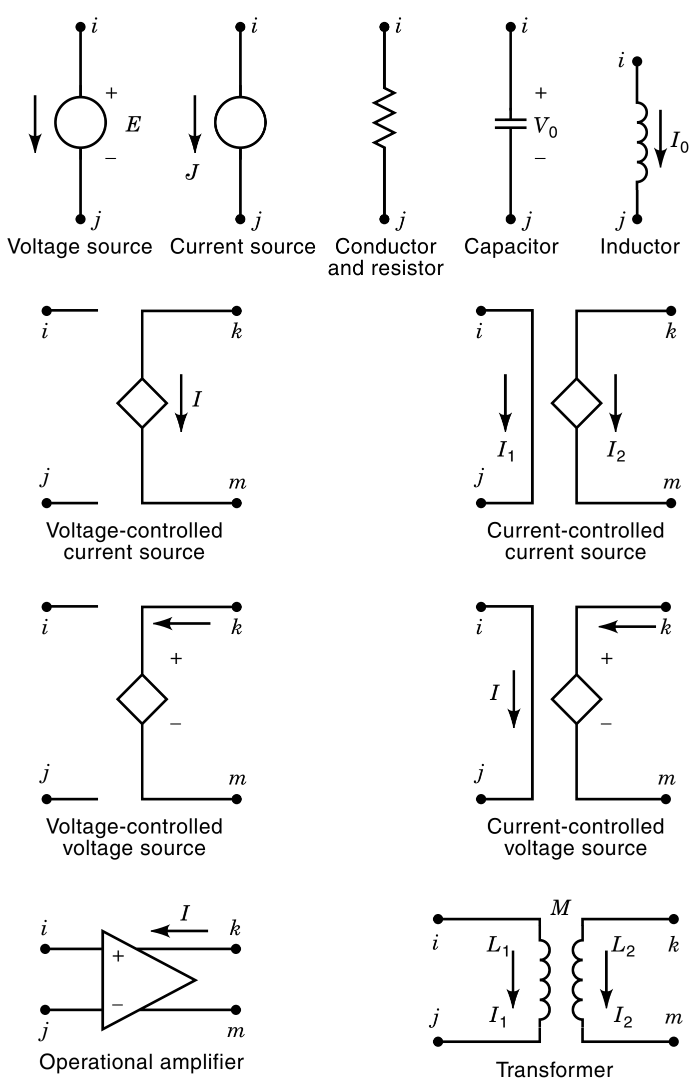

Figure 1. Elements and their symbols.

rent is *always* from the + to the − sign, and this applies to the voltage source as well. If in the application the current actually flows the other way, it is assigned the minus sign. This unified way of defining currents is maintained throughout. It is advantageous to distinguish the voltage supplied by the voltage source from the general voltages in the network, and we will use the letter E. Considering nodal voltages with respect to ground, the voltage source properties are described by

$$V_i - V_j = E \tag{1}$$

An ideal voltage source theoretically maintains its voltage across its terminals no matter what other elements are connected to it. This should be true even if it is a short circuit, an impossible situation. All actual voltage sources have a resistance *in series,* called the internal resistance. If $E = 0$, the ideal voltage source behaves as a short circuit.

Ideal current source is another fundamental source. Its symbol is in Fig. 1. Similarly as in the case of the voltage source, this element theoretically maintains its current no matter what is connected to it. This should theoretically be valid even for an open circuit, another impossible case. Every practical current source will have an internal resistor in parallel to it. Similarly to the above, it is advantageous to distinguish in writing the current delivered by the current source from the other currents in the network. We will use the letter J. If $J = 0$, then the element becomes an open circuit.

Resistor is the most common element. It is usually denoted by the letter R, and its resistance is measured in ohms. Inverted value of the resistance is called the conductance. It is usually denoted by the letter $G = 1/R$ and is measured in siemens. The symbol for the resistor is in Fig. 1. For the purposes of network analysis, it is convenient to consider nodal voltages, measured with respect to ground. The current through the resistor is expressed in terms of nodal voltages by the equation

$$I = G(V_i - V_j) \tag{2}$$

In our considerations we will always assume that the resistor does not change with time. It may change with the current flowing through it or with the voltage across it, and in such a case the resistor is nonlinear. If the value of R is independent of all external influences, then the resistor is linear. In practice, some nonlinearity is always present, but very often we take advantage of the linearity assumption because it greatly simplifies all mathematical steps.

Capacitor is one of two fundamental elements whose behavior depends on the derivative with respect to time. It is usually denoted by the letter C. We will skip many details and only state here that it can hold a charge, usually denoted by the letter Q. The current flowing through the capacitor is defined by

$$I = \frac{dQ(V)}{dt} \tag{3}$$

If the capacitor is linear, the expression simplifies to

$$I = C\frac{dV_i(t) - dV_j(t)}{dt} \tag{4}$$

The capacitance is measured in farads. In Fig. 1, V_0 is the initial voltage on the capacitor.

Inductor is the other element whose behavior depends on the derivative with respect to time. It is usually denoted by the letter L, and its symbol is in Fig. 1. If a current flows through the inductor, a flux Φ is formed and the voltage across the inductor is defined by

$$V_i - V_j = \frac{d\Phi(I)}{dt} \tag{5}$$

In Fig. 1, I_0 is the initial current flowing through the inductor. If the inductor is linear, then its inductance is measured in henry and Eq. (5) simplifies to

$$V_i - V_j = L\frac{dI(t)}{dt} \tag{6}$$

Dependent sources form another set of important elements. Altogether we have four dependent sources:

Voltage-controlled current source (VC) measures a voltage somewhere in the network and adjusts the current at some other terminals. Its symbol is in Fig. 1, and in terms of nodal voltages its performance is expressed by the equation

$$I = g(V_i - V_j) \tag{7}$$

The constant g is called the *transconductance,* is measured in siemens, and the current flows from the terminal k to terminal m. Note that if the difference of the voltages is zero, the current will be zero and the source part will become an open circuit.

Current-controlled current source (CC) measures the current flowing through the short circuit between terminals i and j and delivers a current flowing from terminal k to terminal m. Its performance is expressed by

$$I_2 = \alpha I_1 \tag{8}$$

where α is a dimensionless constant.

Voltage-controlled voltage source (VV) measures a voltage between terminals i and j and forms a voltage source between terminals k and m. In terms of nodal voltages, its performance is expressed by

$$V_k - V_m = \mu(V_i - V_j) \tag{9}$$

where μ, the amplification factor, is a dimensionless constant.

Current-controlled voltage source (CV) measures a current between i and j and delivers voltage between terminals k and m. In terms of nodal voltages, its performance is expresed by the equation

$$V_k - V_m = rI \tag{10}$$

where r is the transresistance measured in ohms. In the above explanations we assumed that the conversion coefficients g, r, α, and μ are constants. They may depend on voltage or current and in such case they will be nonlinear functions of the controlling variable.

Many more elements can be defined, but the above are fundamental and all other elements can somehow be referred to these ones. It is convenient, however, to introduce two more ideal elements.

Operational amplifier (OP) is a voltage-controlled voltage source with an infinitely large gain μ. Its symbol is given in Fig. 1. Often the terminal m is internally grounded and then the symbol has only the terminal k, with the line starting from the tip of the triangle. Using Eq. (9), first divide the equation by μ and then let $\mu \to \infty$. This results in $V_i - V_j = 0$, or $V_i = V_j$. In other words, when analyzing a network with an ideal operational amplifier, then instead of performing similar operations as just described, we simply let both input terminals have the same voltage with respect to ground.

Transformer in its technically realizable form is an element made of two (or more) closely placed inductors, sometimes constructed on a ferromagnetic core. If only two linear inductors are present, then the transformer is described by two equations

$$\begin{aligned} V_i - V_j &= L_1\frac{dI_1}{dt} + M\frac{dI_2}{dt} \\ V_k - V_m &= M\frac{dI_1}{dt} + L_2\frac{dI_2}{dt} \end{aligned} \tag{11}$$

where L_1 and L_2 are the primary and secondary inductors and M is the mutual inductance. For additional information the reader is referred to any introductory book on network theory—for instance, Ref. 1.

KIRCHHOFF'S LAWS

For a systematic writing of equations we need Kirchhoff's laws and some additional rules.

The first rule states that a current is positive when it flows *away* from a node. This is in addition to the previous rule that positive current flows from $+$ to $-$. Thus if we consider the independent voltage source from Fig. 1, in the equations the current at the $+$ sign will be taken as positive, while the current flowing into the $-$ node will be taken as negative.

Kirchhoff discovered two fundamental laws: one for currents, KCL, and one for voltages, KVL.

KCL states that the sum of currents flowing *away* from any node is equal to zero. This means that some currents will flow from the node (and have positive signs), while others will flow to the node (and have negative signs).

KVL states that the sum of voltages around any closed loop is equal to zero.

KCL has yet another definition which we will need later. Assume that we pull two sections of a network apart and consider only the wires which connect them. If we take the currents flowing in the connecting wires from left to right as positive and the others as negative, then this form of KCL states that the sum of currents in these wires will be zero. Rather unfortunately, those who introduced this theory gave it the name *cut.* The cut is of course only in our mind, nothing is changed in the network.

We will explain these laws in more detail in the following sections, but some additional notation has to be understood. In the section entitled "Basic Concepts and Definitions" we stated that if the capacitor is linear, then its current is the derivative of the voltage across it with respect to time, multiplied by the constant C. The dual was stated for a linear inductor. As will be explained later, analysis of nonlinear networks is always done by repeated solutions of linearized approximations. It is thus advantageous to study first linear networks, which we will do here.

Linear networks can be looked upon from many points of view. We can seek time-domain solutions, like in nonlinear networks. In addition, we can apply frequency-domain analysis and find absolute value and phase for a sinusoidal input signal in steady state. To give the reader an easy reference to subjects in other chapters, we will introduce a special symbol for the derivative:

$$s \to \frac{d}{dt} \tag{12}$$

If the network is linear, then s is also the Laplace transform operator, used in frequency domain analysis. In time domain it is the symbol for the derivative. Consider a linear capacitor for which $I = sC(V_i - V_j)$. In frequency domain the operator s will be attached to the constant C. In time domain it will be attached to the voltages, to indicate their derivatives. With these introductory explanations we can now turn to the methods for setting up network equations.

NODAL AND LOOP EQUATIONS

Nodal and loop equations are the simplest methods to write network equations. They are the subject of every fundamental course on network analysis. The methods are not general and are suitable only for hand calculations and small networks. We will start with the more important nodal formulation which is based on KCL.

Consider the network in Fig. 2. The bottom line represents ground. The voltages V_1 and V_2 are nodal voltages, measured with respect to ground. The network has all elements which can be used in nodal analysis without some additional steps. They are the current source, capacitor, conductance, and the voltage-controlled current source, VC. Note that in nodal formulation it is advantageous to work with conductances and not with resistances.

When writing the sum of currents for any node, we do not know anything about the voltages and we are free to *think* that this particular node is the most positive one. This means that all currents through passive elements must flow away from the node under consideration. Using the rules about the signs of currents we write for node 1

$$G_1V_1 + sC(V_1 - V_2) - J = 0$$

and for node 2

$$-sC(V_1 - V_2) + G_2V_2 - g(V_1 - V_2) = 0$$

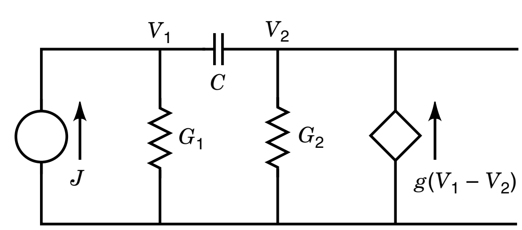

Figure 2. Network for nodal analysis.

In mathematics we normally place known values on the right and collect terms multiplied by the same variable. This leads to

$$(G_1 + sC)V_1 - sCV_2 = J$$
$$(-sC - g)V_1 + (G_2 + sC + g)V_2 = 0$$

Equations describing linear networks can be always put into matrix form:

$$\begin{bmatrix} (G_1 + sC) & -sC \\ (-sC - g) & (G_2 + sC + g) \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} J \\ 0 \end{bmatrix}$$

If we assign unit value to each element except $g = 2$, the matrix reduces to

$$\begin{bmatrix} 1+s & -s \\ -2-s & 3+s \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} J \\ 0 \end{bmatrix}$$

It is solved by any available method. If we were interested in the output, the result would be $V_2 = J(2 + s)/(3 + 2s)$. For frequency-domain analysis we substitute $s = j\omega$ and let $J = 1$. Repeating for a number of frequencies will provide *frequency-domain* response of the network. In time domain the operator s will be replaced by a numerical expression representing the derivative.

Loop equations are based on KVL. It is a method which can be used *only* for planar networks: We must be able to draw the network on a paper without any element crossing another element. The concept of ground is not needed in this formulation. The network in Fig. 3 contains all elements which can be used in this formulation without additional steps. In the figure we indicated fictitious loop currents, circulating in each loop. Note that through the resistor R_2 flow two currents in opposite directions. When writing the sum of voltages around the loop, we consider its circulating current as positive. This leads to the equation

$$R_1I_1 + R_2(I_1 - I_2) - E = 0$$

where our assigned current flows through the source from $-$ to $+$ and thus its contribution must be taken negatively. For the second loop we will consider I_2 as positive and we write the sum of voltages

$$R_2(I_2 - I_1) + sLI_2 + r(I_1 - I_2) = 0$$

The current through CV flows from $+$ to $-$ and thus the voltage it contributes is taken as positive. Transferring E to the right and collecting terms we obtain

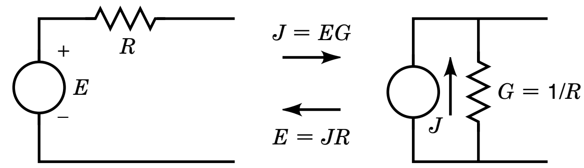

Figure 4. Thevenin–Norton theorem.

$$(R_1 + R_2)I_1 - R_2I_2 = E$$
$$(-R_2 + r)I_1 + (R_2 + sL - r)I_2 = 0$$

In matrix form this is

$$\begin{bmatrix} (R_1 + R_2) & -R_2 \\ (-R_2 + r) & (R_2 + sL - r) \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} E \\ 0 \end{bmatrix}$$

Selecting $r = 2$ and assigning unit values to all other elements reduces the equation to

$$\begin{bmatrix} 2 & -1 \\ 1 & -1+s \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} E \\ 0 \end{bmatrix}$$

Suppose now that we would like to analyze the first network by the loop method and the second by the nodal method. It is not possible directly and we must apply some transformations before we can proceed.

The Thevenin–Norton transformation states that a voltage source with a resistor in series can be transformed into a current source with the same resistor in parallel. As far as the other parts of the network are concerned, there will be no difference. The transformation is schematically shown in Fig. 4. The sources are coupled by a law which looks like Ohm's law, but applies to the sources:

$$E = RJ \quad \text{or} \quad J = GE \tag{13}$$

As indicated in Fig. 4, we can go with the transformation in either direction.

As an example, we will transform the current sources in Fig. 2 into voltage sources. The transformed network is in Fig. 5. To proceed, we must express the dependent source in terms of the unknown current, $V_1 - V_2 = I/sC$. Afterwards we must find the voltage V_2 as a sum of the voltages delivered by the dependent source plus the voltage across the resistor R_2. The result is, of course, the same as before, but we had to go through a number of additional steps. Similar problems

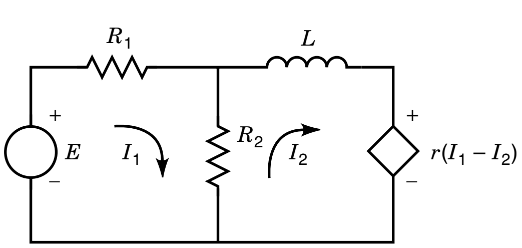

Figure 3. Network for loop analysis.

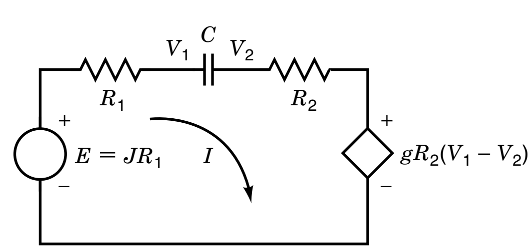

Figure 5. The network in Fig. 2, transformed for loop analysis.

would be with the transformation of network in Fig. 3 for analysis by the nodal method. It can be seen that the transformations make the whole process fairly complicated and unsuitable for computerized programming. Other methods had to be invented: the state variables, the tableau, and the modified nodal formulations. To proceed with their explanations we will need some fundamental concepts of the graph theory, but before that we discuss methods for the solution of systems of equations. Details of the various transformations can be found in Ref. 1.

SOLUTIONS OF NETWORK EQUATIONS

Only the smallest networks can be solved by hand; all practical networks must be solved by computer, and this leads us to the methods used in such solutions.

Equations describing linear networks are expressed in the form of a matrix equation

$$\mathbf{TX} = \mathbf{W} \tag{14}$$

Here $\mathbf{T}$ is the system matrix, $\mathbf{X}$ are the system variables, in most cases voltages and/or currents, and $\mathbf{W}$ denotes the sources. Our examples in Figs. 2 and 3 were brought to this final matrix form.

Networks with up to 1000 nodes are almost always solved by a process called "LU" or "triangular" decomposition. Many books describe this process, and the reader is referred to other sources—for instance, Refs. 2–5. However, because we will be referring to it in the following, at least some general concepts will be given.

Suppose that we manage to decompose the matrix $\mathbf{T}$ into the product of two matrices, $\mathbf{T} = \mathbf{LU}$ where the matrix $\mathbf{L}$ is lower triangular, with all the entries above the main diagonal being zero. The $\mathbf{U}$ matrix has all entries below the main diagonal zero and, in addition, all entries on the main diagonal are units. The system in Eq. (15) is rewritten as

$$\mathbf{LUX} = \mathbf{W} \tag{15}$$

Suppose that we now introduce a new definition,

$$\mathbf{UX} = \mathbf{Z} \tag{16}$$

This cannot be solved yet, but inserting into Eq. (15) we can write

$$\mathbf{LZ} = \mathbf{W} \tag{17}$$

Because the matrix is triangular, a simple process, called *forward substitution,* can be used to find $\mathbf{Z}$. Once this is known, we go back to Eq. (16) and find $\mathbf{X}$ by a similarly simple process, called *back substitution.* The important point of this process is that the decomposition into the $\mathbf{LU}$ matrices costs $n^3/3$ multiplication/division operations, while the forward-back substitution costs only n^2 operations, n being the size of the matrix. If the right-hand side changes, only new forward and back substitution is needed, and the $\mathbf{LU}$ decomposed matrix is re-used.

Almost all larger networks have system matrices with many zeros, and a special processing, called *sparse matrix decomposition,* can be used. Computer codes may be fairly complex, but sparse matrix solutions are done with a cost approximately proportional to n and not n^3. In fact, the discovery of sparse matrix processing made it possible to write practical programs for the analysis of quite large networks.

All network solutions eventually reduce to the solution of a system of linear equations. In frequency domain, s is substituted by $j\omega$ and programming is in complex. In time domain, the derivatives are replaced by an approximation which changes the system of algebraic and differential equations into a system of algebraic equations only.

Nonlinear networks are solved by a method leading to a repeated solution of linear approximations. The method is known as the "Newton–Raphson iteration," and we will explain it with a set of two equations in two unknowns:

$$\begin{aligned} f_1(x_1, x_2) &= 0 \\ f_2(x_1, x_2) &= 0 \end{aligned} \tag{18}$$

Expand each equation into a Taylor series about the point $x_1 + \Delta x_1$ and $x_2 + \Delta x_2$. The expansion is

$$\begin{aligned} f_1(x_1, x_2) + \frac{\partial f_1}{\partial x_1}\Delta x_1 + \frac{\partial f_1}{\partial x_2}\Delta x_2 + \text{higher terms} &= 0 \\ f_2(x_1, x_2) + \frac{\partial f_2}{\partial x_1}\Delta x_1 + \frac{\partial f_2}{\partial x_2}\Delta x_2 + \text{higher terms} &= 0 \end{aligned} \tag{19}$$

If we neglect the higher terms, we can rewrite these equations as

$$\begin{aligned} \frac{\partial f_1}{\partial x_1}\Delta x_1 + \frac{\partial f_1}{\partial x_2}\Delta x_2 &= -f_1(x_1, x_3) \\ \frac{\partial f_2}{\partial x_1}\Delta x_1 + \frac{\partial f_2}{\partial x_2}\Delta x_2 &= -f_2(x_1, x_2) \end{aligned} \tag{20}$$

This is a set of linear equations which can be written in matrix form:

$$\begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} \\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = \begin{bmatrix} -f_1(x_1, x_2) \\ -f_2(x_1, x_2) \end{bmatrix} \tag{21}$$

The matrix on the left is called the *Jacobian;* we will denote it by the letter $\mathbf{M}$, the vector of unknowns by $\Delta \boldsymbol{X}$, and the right-hand side by $\mathbf{f}$,

$$\mathbf{M}\Delta \boldsymbol{X} = -\mathbf{f}$$

Since we neglected higher terms, this is not a final solution. We denote the kth step with the superscript k and rewrite the process as two equations:

$$\begin{aligned} \mathbf{M}^k \Delta \boldsymbol{X}^k &= -\mathbf{f}^k \\ \mathbf{X}^{k+1} &= \mathbf{X}^k + \Delta \boldsymbol{X}^k \end{aligned} \tag{22}$$

In the first one we apply LU decomposition to the matrix and find $\Delta \boldsymbol{X}$. The second equation finds new $\mathbf{X}$, closer to the correct solution. With it we go back to the first equation, and so on. If the process converges, the Δx_i will eventually become very small and we stop the iteration.

GRAPHS

Network formulations suitable for computer applications require at least a few graph concepts, and we devote this section to the subject (2).

Graph theory attempts to extract basic properties of a network without giving any details about the network elements. An element is replaced by a line in which the assumed direction of the current is indicted by an arrow. For passive elements we are free to select this direction. Sources have their current directions given by the previous rules, and the arrow must agree with them: For the voltage source the arrow will go from + to −, for the current source it will be the direction indicated at the current source symbol.

Consider the graph in Fig. 6 representing some network with six elements, replaced by directed graphs. Nodes are indicated by numbers in circles. For nodes 1, 2, and 3 we write three KCL equations:

$$
\begin{aligned}
-I_1 + I_4 + I_6 &= 0\\
-I_2 - I_4 + I_5 &= 0\\
-I_3 - I_5 - I_6 &= 0
\end{aligned}
$$

They can be written in matrix form:

$$
\begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 1\\ 0 & -1 & 0 & -1 & 1 & 0\\ 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix}
\begin{bmatrix} I_1\\ I_2\\ I_3\\ I_4\\ I_5\\ I_6 \end{bmatrix} = \begin{bmatrix} 0\\ 0\\ 0 \end{bmatrix}
$$

or

$$\mathbf{AI} = \mathbf{0} \tag{23}$$

The matrix **A** is called the *incidence matrix.* It also couples nodal voltages, V_n, to the voltages across the elements, V_b, by the equation

$$\mathbf{V}_b = \mathbf{A}^{\mathrm{T}}\mathbf{V}_n \tag{24}$$

where the superscript T denotes the transpose of **A**. Equations (23) and (24) will be needed to explain the tableau formulation.

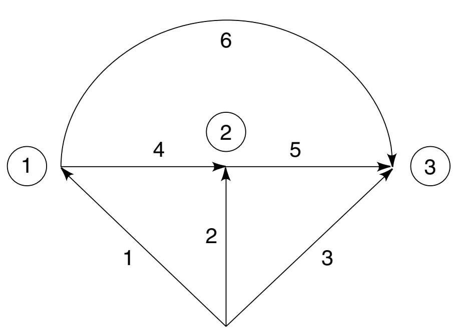

Figure 6. Graph for incidence matrix.

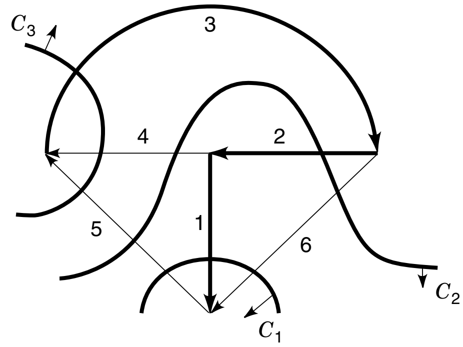

Figure 7. Graph with a tree.

To work with another formulation method, the *state variables,* we need additional graph concepts of the tree and cotree. Consider Fig. 7, where we have used the same graph but where we selected a *tree,* indicated by bold lines. A tree is such a selection of lines of the graph which connects all nodes but which does not close a loop. Directions of the arrows are again arbitrary, except for lines representing sources. The thin lines represent the *co-tree.* The figure also shows three cuts, C_i. Each cut goes through *only one* line of the tree and as many lines of the co-tree as necessary to separate the network into two parts. If we sum the currents in these "cuts" but taking the direction of the tree line as positive, we end up with the following set of equations, written in the sequence of the cuts C_i:

$$
\begin{aligned}
I_1 - I_5 + I_6 &= 0\\
I_2 - I_4 - I_5 + I_6 &= 0\\
I_3 - I_4 - I_5 &= 0
\end{aligned}
$$

In matrix form

$$
\begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 1\\ 0 & 1 & 0 & -1 & -1 & 1\\ 0 & 0 & 1 & -1 & -1 & 0 \end{bmatrix}
\begin{bmatrix} I_1\\ I_2\\ I_3\\ I_4\\ I_5\\ I_6 \end{bmatrix} = \begin{bmatrix} 0\\ 0\\ 0 \end{bmatrix}
$$

or briefly

$$\mathbf{QI} = \mathbf{0}$$

In selecting the tree we used the following sequence of rules:

1. Assign orientations to all lines.
2. Select a tree.
3. Assign consecutive integers starting from 1 to the lines of the tree and continue numbering the lines of the co-tree.

If we follow these three steps, it will *always* be true that the matrix **Q** will have in the left partition a unit matrix, followed by a partition describing directions of the co-tree:

$$\mathbf{Q} = [\mathbf{1} \mid \mathbf{Q_c}]$$

Similarly as in the case of the incidence matrix, tree (subscript t) and co-tree (subscript c) voltages and currents can be related by means of the $\mathbf{Q}_c$ matrix:

$$\mathbf{I}_t = -\mathbf{Q}_c\mathbf{I}_c \tag{25}$$

$$\mathbf{V}_c = \mathbf{Q}_c^T\mathbf{V}_t \tag{26}$$

Complete derivations of these equations can be found in Ref. 2. We will need Eqs. (25) and (26) when we speak about the state variable formulation.

STATE VARIABLES

Historically, "state variables" were the first method used to write equations for larger networks. The idea was to reduce the system to a set of first-order differential equations. Over the years, many attempts were made to write a general analysis program based on state variables, but none of them succeeded. As a result, this method can be used for relatively small networks, mostly for manual solution. It is useful for theoretical studies, and it is still applied in some disciplines.

A system of first-order differential equations can be written in matrix form as

$$\mathbf{X}' = \mathbf{AX} + \mathbf{BW} \tag{27}$$

On the left is the derivative of the $\boldsymbol{X}$ vector, composed of voltages and currents, and $\mathbf{W}$ describes the sources. This equation is usually accompanied by another matrix equation for the outputs:

$$\mathbf{Y} = \mathbf{CX} + \mathbf{DW} \tag{28}$$

In our explanations we will use only Eq. (27). The graph theory and the $\mathbf{Q}$ matrix which were explained in the previous section related tree and co-tree voltages and currents by means of Eqs. (25) and (26). Importance of the equations lies in the fact that independent variables are tree voltages and co-tree currents. Because in (27) we need the derivatives, consider the expression $I = C[dV(t)/dt]$. It indicates that we should retain capacitor voltages as independent variables, which is helped by taking capacitors into the tree. Dually, derivatives of currents appear in $V = L[dI(t)/dt]$. The derivatives should be retained, which is helped by placing inductors into the co-tree. In practical networks we often experience situations where capacitors form a loop and thus not all can be taken into the tree. A dual situation may also happen with the inductors.

Independent voltage sources also require special attention: Their voltages are known and thus cannot be considered as dependent variables. This means that the voltage sources graph lines *must* be taken into the tree. Dually, current sources must be taken into the co-tree.

We will demonstrate some of the problems on the small network and its graph in Fig. 8. Only two capacitors can be taken into the tree, and the inductor is in the co-tree. Using the graph we set up the $\mathbf{Q}$ matrix as explained in the previous section:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 & 1 \end{bmatrix}$$

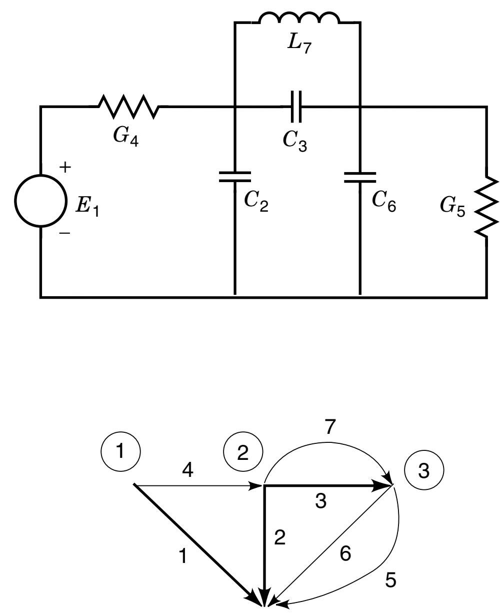

Figure 8. Example for state variables.

The last four columns create the matrix $\mathbf{Q}_c$. In the next step we take Eqs. (25) and (26) and put them into one matrix equation:

$$\begin{bmatrix} \mathbf{0} & -\mathbf{Q}_c \\ \mathbf{Q}_c^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V}_t \\ \mathbf{I}_c \end{bmatrix} = \begin{bmatrix} \mathbf{I}_t \\ \mathbf{V}_c \end{bmatrix}$$

Writing the matrix and inserting into the column vectors expressions describing properties of the elements we obtain one matrix equation:

$$\begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} E_1 \\ V_2 \\ V_3 \\ G_4V_4 \\ G_5V_5 \\ sC_6V_6 \\ I_7 \end{bmatrix} = \begin{bmatrix} I_1 \\ sC_2V_2 \\ sC_3V_3 \\ V_4 \\ V_5 \\ V_6 \\ sL_7I_7 \end{bmatrix}$$

As an intermediate step we must eliminate all variables except V_2, V_3, and I_7, namely, the tree voltages and co-tree currents. After a number of steps which we omit, we get the preliminary result

$$\begin{bmatrix} C_2 + C_6 & -C_6 & 0 \\ -C_6 & C_3 + C_6 & 0 \\ 0 & 0 & L_7 \end{bmatrix} \begin{bmatrix} dV_2/dt \\ dV_3/dt \\ dI_7/dt \end{bmatrix} = \begin{bmatrix} -G_4 - G_5 & G_5 & 0 \\ G_5 & -G_5 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_2 \\ V_3 \\ I_7 \end{bmatrix} + \begin{bmatrix} G_4E \\ 0 \\ 0 \end{bmatrix}$$

This is still not the state variable form, because there is a matrix on the left. Our matrix happens to be nonsingular and could be inverted to get the form Eq. (27). In many situations the matrix on the left turns out to be singular and additional eliminations are necessary. Our example did not have any dependent sources. If present, they contribute with algebraic equations which must also be eliminated. As can be seen, state variable formulation creates so many problems that it was effectively abandoned in computer applications. Its usefulness seems to be in theoretical studies of relatively small problems. We do not recommend its use, but we felt that some explanations are necessary. Details on state variables can be found in many books—for instance, in Refs. 6 and 7.

TABLEAU

This method is more modern than state variables (8), but it still has problems. In some way it does exactly the opposite to the state variables: Instead of eliminating as many equations as possible, the tableau retains all of them. This would not be a good idea, except for the fact that the equations are very sparse and sparse matrix methods can be used. Recall that the price of sparse matrices processing is approximately proportional to n instead of n^3, as is the case for full matrices. However, the sparse matrix solver turns out to be very complex. Unless the reader has access to a solver for tableau, we suggest not to use this method, but we explain at least its principles.

In the section on graphs we introduced the concept of the incidence matrix **A**, with Eqs. (23) and (24). To complete network description, we need a general expression suitable for any element. It turns out that this is possible by writing

$$\mathbf{Y}_b\mathbf{V}_b + \mathbf{Z}_b\mathbf{I}_b = \mathbf{W}_b \tag{29}$$

To show that this is true, for instance, for the voltage source defined by $V_b = E$, select $Y_b = 1$, $Z_b = 0$, and $W_b = E$. Four terminal networks, like the dependent sources, are represented by two equations and their graphs must have two graph lines: one for the input and one for the output. The reader should test validity of these statements for all networks in Fig. 1. Details can be found in Ref. 2.

Equations (23), (24), and (29) can be collected into one matrix equation:

$$\begin{bmatrix} \mathbf{1} & \mathbf{0} & -\mathbf{A}^T \\ \mathbf{Y}_b & \mathbf{Z}_b & \mathbf{0} \\ 0 & \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_b \\ \mathbf{I}_b \\ \mathbf{V}_n \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{W}_b \\ \mathbf{0} \end{bmatrix} \tag{30}$$

This is the tableau formulation. Note how simple it is to write it once we have set up the incidence matrix.

$$\left[\begin{array}{cccc|cccc|cc} 1 & & & & & & & & 1 & 0 \\ & 1 & & & & \mathbf{0} & & & -1 & 0 \\ & & 1 & & & & & & -1 & 1 \\ & & & 1 & & & & & 0 & -1 \\ \hline 0 & & & & 1 & & & & & \\ & 1 & & & & -R_2 & & & & \mathbf{0} \\ & & sC_3 & & & & -1 & & & \\ & & & G_4 & & & & -1 & & \\ \hline & \mathbf{0} & & & -1 & 1 & 1 & 0 & & \mathbf{0} \\ & & & & 0 & 0 & -1 & 1 & & \end{array}\right] \left[\begin{array}{c} V_1 \\ V_2 \\ V_3 \\ V_4 \\ \hline I_1 \\ I_2 \\ I_3 \\ I_4 \\ \hline V_{n1} \\ V_{n2} \end{array}\right] = \left[\begin{array}{c} \\ \mathbf{0} \\ \\ \\ \hline J_1 \\ 0 \\ 0 \\ 0 \\ \hline \mathbf{0} \\ \\ \end{array}\right]$$

Figure 10. Tableau matrix equation for example in Fig. 9.

We will use one example to indicate how the system matrix is set up. Consider the network in Fig. 9 with its graph. The incidence matrix is

$$\mathbf{A} = \begin{bmatrix} -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

The network has four branch voltages, V_b, four branch currents, I_b, and two nodal voltages, V_n. The elements are described by the equations $I_1 = J_1$, $V_2 = R_2I_2$, $V_3 = sC_3V_3$, and $V_4 = G_4V_4$. The system will have the size 10. Filling the entries we obtain the tableau matrix shown in Fig. 10. The matrix has 100 entries, but only 21 are nonzero and only three would be real numbers. Had we used nodal formulation, the network would be described by two equations only. This clearly shows that the tableau is useful *only* on computers and *only* if an appropriate sparse matrix solver is available.

MODIFIED NODAL FORMULATION

Modified nodal formulation (9) is the method used in practically all modern simulators. It is based on nodal formulation to which additional equations are added as needed. Recall from the section entitled "Solutions of Network Equations" that only four elements can be used in nodal formulation directly: current source, capacitor, conductor, and voltage-controlled current source (VC). Also recall that Thevenin–Norton transformations can help, but the steps become very difficult for programming.

To clarify the ideas consider the network in Fig. 11 but think of the inductor as replaced by the current which flows

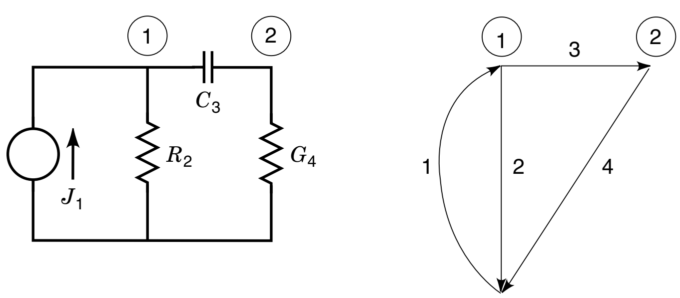

Figure 9. Example for tableau.

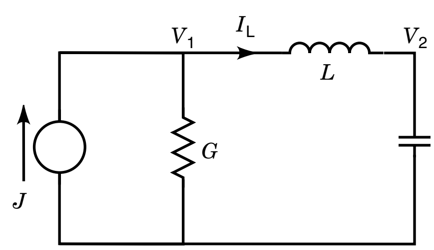

Figure 11. First example for modified nodal formulation.

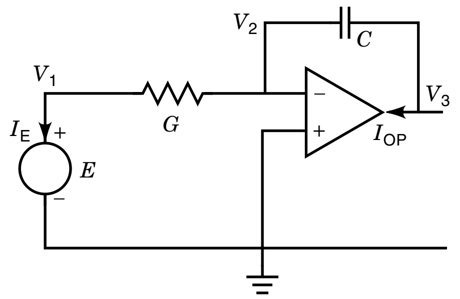

Figure 12. Second example for modified nodal formulation.

through it. Using KCL for node 1 we write

$$GV_1 + I_L - J = 0$$

Since the current I_L flows away from node 1, it is taken positive. At node 2 it must be taken negative and KCL for this node leads to

$$-I_L + sCV_2 = 0$$

So far we have two equations and three unknowns. The set must be completed with an equation describing properties of the inductor:

$$V_1 - V_2 - sLI_L = 0$$

The three equations can be put into a matrix form:

$$\begin{bmatrix} G & 0 & 1 \\ 0 & sC & -1 \\ 1 & -1 & -sL \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_L \end{bmatrix} = \begin{bmatrix} J \\ 0 \\ 0 \end{bmatrix} \tag{31}$$

This is the modified nodal formulation. Since multiplication by s represents the derivative with respect to time, our steps resulted in a system with one algebraic and two differential equations.

Consider next the network in Fig. 12. It has one voltage source and one operational amplifier, elements which cannot be taken into nodal formulation. Similarly as above we first replace them with their currents, as shown, and apply KCL to the three ungrounded nodes:

$$\begin{aligned} G(V_1 - V_2) + I_E &= 0 \\ -GV_1 + (G + sC)V_2 - sCV_3 &= 0 \\ -sCV_2 + sCV_3 + I_{OP} &= 0 \end{aligned}$$

To this set we append equations describing properties of the two elements. For the voltage source $V_1 = E$. For the operational amplifier we know that the two input terminals are at the same potential. Since one of them is grounded, the second will be at zero potential and $-V_2 = 0$. Adding these equations to the above set we get in matrix form

$$\begin{bmatrix} G & -G & 0 & 1 & 0 \\ -G & G+sC & -sC & 0 & 0 \\ 0 & -sC & sC & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ I_I \\ I_{OP} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ E \\ 0 \end{bmatrix} \tag{32}$$

The two networks introduced the principles we use: First we replace all elements which cannot be taken into nodal formulation by their currents and write KCL. Then we append equations describing properties of the elements. Let us return to the voltage source. In a general case $V_i - V_j = E$ and this must be added to the previous set of equations. In addition, the current flowing through the voltage source must also be added at node i and subtracted at node j. This is symbolically summarized in the stamp shown in Fig. 13. On the left the rows are marked by i and j. They correspond to the nodes where the currents are added or subtracted. Above the stamp the letters i and j are subscripts of the nodal voltages which multiply the columns. The units appearing in the stamp are thus multiplied by either a voltage or a current, also indicated above the stamp. Following similar considerations we can derive stamps for all the elements introduced in the section entitled "Basic Concepts and Definitions." They are collected in Fig. 13.

The starting matrix is always the nodal matrix; its size is n, the number of ungrounded nodes. Afterwards additional row(s) and column(s) are added one by one, as we keep adding the various elements. This was actually done when we considered the network in Fig. 12. First we added the row and column for the voltage source, and afterwards we added to this already increased matrix the row and column for the operational amplifier.

A reader wishing to learn more about this formulation is referred to Refs. 1 and 2.

SIMPLE INTEGRATION METHODS

Methods for integration of differential equations were developed much earlier than programs for simulation, and first attempts of computerized network simulations were directed to the use of known integration procedures. State variables were developed because integration of first-order differential equations was available in the Runge–Kutta routines. When problems were encountered, new methods for integration were developed.

The majority of integration methods are polynomial approximations of various orders. In this section we will consider three simplest formulas for numerical integration: the forward Euler, the backward Euler, and the trapezoidal (2). They are practical methods, used in commercial simulation packages.

Consider a given differential equation

$$x' = f(x, t) \tag{33}$$

where x' replaces dx/dt for simplicity. Let the initial value x_0 be known at $t = 0$. The derivative x'_0 can be evaluated by inserting into Eq. (33). The simplest formula to predict the value x_1 at $t = h$ can be derived by inspecting Fig. 14:

$$x_1 = x_0 + hx'_0 \tag{34}$$

All terms on the right are available: The formula is explicit and belongs to the class of predictors. Its name is *forward Euler.* Another formula, the *backward Euler,* makes the estimate differently:

$$x_1 = x_0 + hx'_1 \tag{35}$$

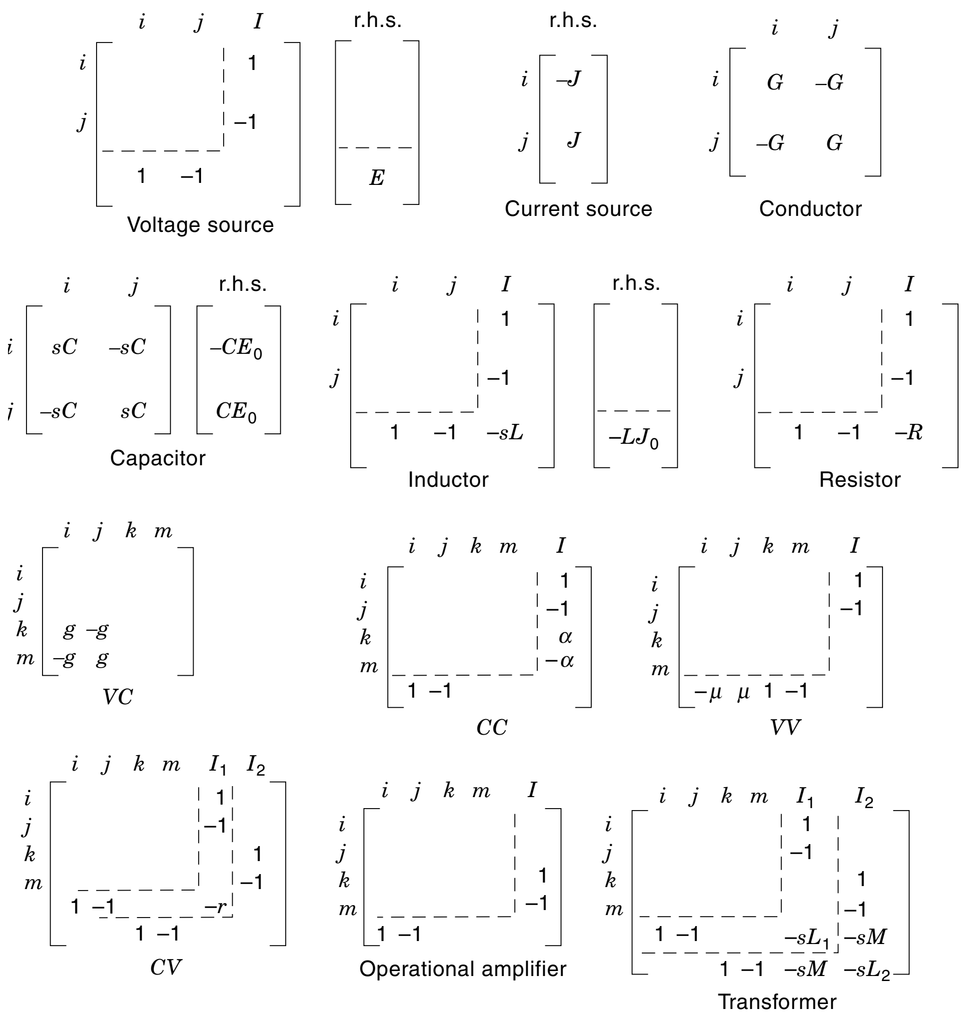

Figure 13. Stamps for the networks in Fig. 1.

Because the right side contains the unknown derivative at $t = h$, this formula is *implicit* and belongs to the class of *correctors*. It would seem that Eq. (34) is better, because it is simpler. Actually, Eq. (35) is much better, but Eq. (34) has its use as well. It can be applied at the beginning of every step to predict the new point x_1. The information is then used in Eq. (33) to find an approximation to x_1', which in turn can be inserted into Eq. (35). Repeating several times between Eq. (33) and Eq. (35) we eventually come to a situation when x_1, substituted into Eq. (33), provides x_1' which we already had.

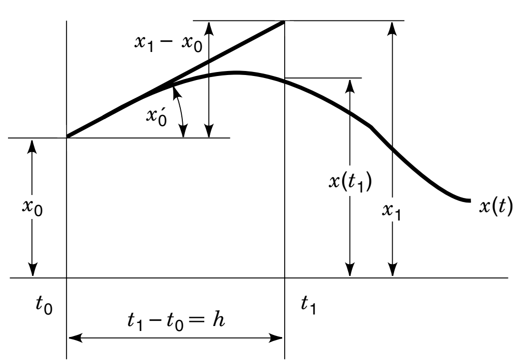

Figure 14. Approximations for forward Euler formula.

The iterations have converged and the process can be repeated for another step to find x_2, and so on. Both Euler formulas match the first derivative and we say that their *order of integration* is one.

If we take the sum of Eq. (33) and Eq. (34), we get another corrector, called *trapezoidal:*

$$x_1 = x_0 + \frac{h}{2}(x_1' + x_0') \tag{36}$$

Its order of integration is two.

Properties of integration methods are generally studied on the simplest differential equation (2,10)

$$x' = \lambda x \tag{37}$$

Its exact solution is

$$x(t) = x_0 e^{\lambda t} \tag{38}$$

The constant λ can be real or complex. Using simple steps (2), it is possible to derive for the three formulas important stabil-

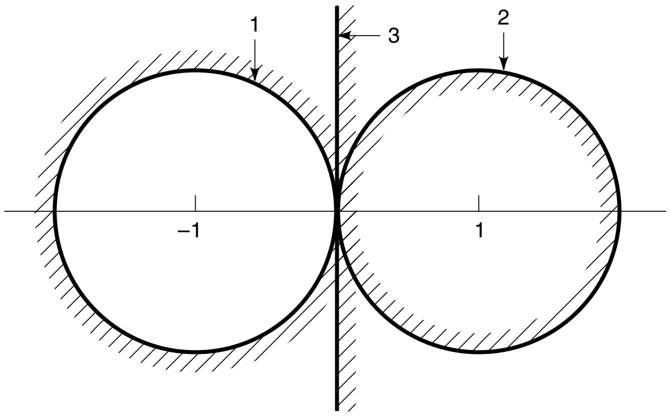

Figure 15. Stability properties of integration formulas. (1) Forward Euler, stable inside; (2) backward Euler, stable outside; (3) trapezoidal, stable in the left half-plane.

ity properties, summarized in Fig. 15. Areas indicated by hatching indicate unstable regions of each formula. Let λ have negative real part, which means that $x(t)$ in Eq. (38) will tend to zero for large t, irrespective of what is its imaginary part.

The left unit circle refers to the forward Euler formula. If the product λh can be plotted inside this circle, then application of the formula will give results which will tend to zero for a large number of steps, similarly as the exact solution. However, if the point falls outside the unit circle, the results by the formula will incorrectly grow with the number of steps. The consequence is that for large absolute value of λ we must choose a small step size h to get the point into the unit circle. Stability, however, does not yet mean accuracy. For that the product λh must fall close to the origin of axes in the figure.

The outside of the right unit circle corresponds to the backward Euler formula and its stable region. It shows that for the negative real part of λ the solution will be always in the stable region, no matter how large a step we take. Again, for accuracy the point should be close to the origin.

The vertical axis is the border between stable and unstable regions of the trapezoidal formula. If the real part of λ is negative, both exact solution and results by the formula will tend to zero for large t. The opposite will be true for the positive real part, which is again correct. This is called "absolute" stability. There exists a proof by Dahlquist that no higher-order formula can be absolutely stable.

ADVANCED INTEGRATION METHODS

Integration methods can be self-starting or can use a number of previous solutions. Self-starting are, for instance, the Runge–Kutta formulas. If previous solutions are used, the formulas are generally known as *multistep.* Many such methods are available, but the only ones used these days are the *backward differentiation formulas* (BDF) (2,10,11).

If the network is linear and has parasitic elements, then its responses will be a weighted sum of functions Eq. (38) with very different λ_i. Such systems are called *stiff.* Should we integrate such a system with the forward Euler method, *every* λ multiplied by h must be pulled into the unit circle to preserve stability of integration. If we use the backward Euler method, we still need to pull the desired poles close to the origin, but the large λ_i will not bother us. Their responses will not be traced, but we in fact do not want them, because their rapid changes do not contribute to the useful function of the network. The BDF formulas behave (roughly speaking) similarly to the backward Euler method (2).

Assume that we have the solution x_n at the instant t_n, as well as a number of previous solutions x_{n-i} at instants t_{n-i}, as sketched in Fig. 16 with three previous solutions. A new step h reaches the instant t_{n+1} where we wish to find the solution. For our explanation we will assume equal integration steps, with BDF formulas collected in Table 1. They are actually polynomials passing through known points and extrapolated to the next time instant. The formulas are used as follows. At the beginning $n = 0$, we know the initial value x_0 and we select h. The zero-order predictor from Table 1 estimates the value of $x_{n+1} = x_1$. With this and the previous point we can use the corrector of order 1. It is, in fact, the already known backward Euler formula.

We have now two possibilities. Either we use always only two last points and in a sequence of steps apply the first-order predictor or corrector. The other possibility is to take more of the already known solutions and use a predictor corrector pair of higher order.

Equal steps are not very practical, although in some cases they are used. If we permit a change of the step in every new evaluation, then the BDF formulas change, as summarized in Table 2. The values z_k in Table 2 are expressed by

$$z_k = \frac{t_{n+1} - t_{n+1-k}}{h} \tag{39}$$

If we use higher-order formulas, z_k must be saved simultaneously with the previous solutions.

The step size influences accuracy and for correct integration we must estimate the error. This is where the importance of the predictor–corrector pair of the same order comes into the picture. It was shown in Refs. 12 and 13 that the error is expressed by

$$E = \frac{h(x_{n+1}^{\text{pred}} - x_{n+1}^{\text{cor}})}{a_0(t_{n+1} - t_{n+1-k})} = \frac{hD}{a_0 T} \tag{40}$$

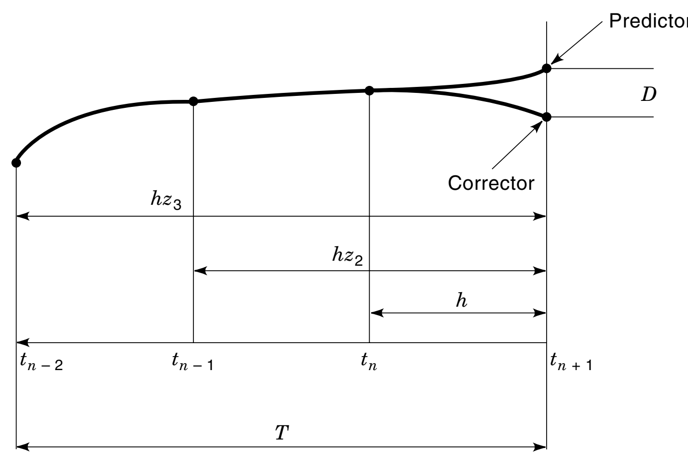

Figure 16. Estimating error from predictor and corrector results.

Table 1. BDF Formulas, Equal Steps

Order	Predictors	Correctors
0	$x_{n+1} = x_n$	$x'_{n+1} = 0$
1	$x_{n+1} = 2x_n - x_{n-1}$	$x'_{n+1} = \frac{1}{h}(x_{n+1} - x_n)$
2	$x_{n+1} = 3x_n - 3x_{n-1} + x_{n-2}$	$x'_{n+1} = \frac{1}{h}\left(\frac{3}{2}x_{n+1} - 2x_n + \frac{1}{2}x_{n-1}\right)$
3	$x_{n+1} = 4x_n - 6x_{n-1} + 4x_{n-2} - x_{n-3}$	$x'_{n+1} = \frac{1}{h}\left(\frac{11}{6}x_{n+1} - 3x_n + \frac{3}{2}x_{n-1} - \frac{1}{3}x_{n-2}\right)$

where k is the number of points taken into consideration, and D and T are shown in Fig. 16. The coefficient a_0 is the coefficient multiplying x_{n+1} in the corrector formula. If the error is large, we reduce the step size. There exist advanced methods on how to adjust the step, all beyond the scope of this contribution, but in many cases the step is simply halved and the process tried again. One can similarly increase the step if the error turns out to be smaller than permitted.

Most commercial packages use only the Euler formulas and the trapezoidal rule, and some also use the second-order BDF formula. Higher-order integration turns out to be efficient only if nonlinearities have several continuous derivatives. Since most transistor models have only the first derivative continuous, there would be no advantage in switching to orders higher than two (14).

LINEAR NETWORKS

Linear networks offer a large number of possibilities how to study them. They can be analyzed in frequency domain and time domain, network functions can be derived, and poles and zeros can be calculated.

Since our modified nodal formulation was explained on linear networks, it is worth mentioning how simple frequency domain analysis is. In frequency domain we calculate how a sinusoidal input signal would be transferred through the network after all transients have died out. Suppose that we have the equations in matrix form, similarly as in the section entitled "Nodal Formulation." Once we have the equations in matrix form, all we do is insert a unit value for the source E or J and substitute in the matrix s by $j\omega$, where $\omega = 2\pi f$ and f is the frequency of interest. The program for LU decomposition must be in complex arithmetic, and the resulting solution variables are complex as well. Absolute value or phase can be obtained from such complex values. Repeating for a number of frequencies, we get the frequency-domain response.

Time-domain solutions are more complicated, but still simple enough when considering backward Euler or trapezoidal formulas. Recall that multiplication by s represents the derivative and write

$$\mathbf{G}\mathbf{X}_{n+1} + \mathbf{C}\mathbf{X}'_{n+1} = \mathbf{W}_{n+1} \tag{41}$$

We added the subscript $n + 1$ to indicate integration steps. In Eq. (41), $\mathbf{G}$ are all entries of the matrix not multiplied by s, and $\mathbf{C}$ are all entries multiplied by s. The backward Euler formula can be similarly expressed by

$$\mathbf{X}'_{n+1} = \frac{1}{h}(\mathbf{X}_{n+1} - \mathbf{X}_n) \tag{42}$$

Inserting into Eq. (41) provides (2)

$$\left(\mathbf{G} + \frac{1}{h}\mathbf{C}\right)\mathbf{X}_{n+1} = \frac{1}{h}\mathbf{C}\mathbf{X}_n + \mathbf{W}_{n+1} \tag{43}$$

On the left is the same matrix as we had before, with s replaced by $1/h$. On the right, the $\mathbf{C}$ matrix is multiplied by the previous result, $\mathbf{X}_n$, and added to the vector of the sources, evaluated at the next time instant. Now suppose that we keep the step size fixed during the whole integration. In such a case the matrix on the left does not change and all we need is one LU decomposition for the entire time-domain calcula-

Table 2. BDF Formulas, Variable Steps

Order	Predictors	Correctors
0	$x_{n+1} = x_n$	$x'_{n+1} = 0$
1	$a_1 = z_2/(z_2 - 1)$ $a_2 = 1/(1 - z_2)$ $x_{n+1} = a_1x_n + a_2x_{n-1}$	$a_0 = 1$ $a_1 = -1$ $x'_{n+1} = \frac{1}{h}(a_0x_{n+1} + a_1x_n)$
2	$D = (z_3 - z_2)(1 + z_2z_3 - z_2 - z_3)$ $a_1 = z_2z_3(z_3 - z_2)/D$ $a_2 = z_3(1 - z_3)/D$ $a_3 = z_2(z_2 - 1)/D$ $x_{n+1} = a_1x_n + a_2x_{n-1} + a_3x_{n-2}$	$D = z_2(z_2 - 1)$ $a_0 = (z_2^2 - 1)/D$ $a_1 = -z_2^2/D$ $a_2 = 1/D$ $x'_{n+1} = \frac{1}{h}(a_0x_{n+1} + a_1x_n + a_2x_{n-1})$

tion. In each step we prepare a new right-hand side and find $\mathbf{X}_{n+1}$ by forward and back substitution. For the example in Fig. 11 this system would be

$$\begin{bmatrix} G & 0 & 1 \\ 0 & C/h & -1 \\ 1 & -1 & -L/h \end{bmatrix} \begin{bmatrix} V_{1,n+1} \\ V_{2,n+1} \\ I_{L,n+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & C/h & 0 \\ 0 & 0 & -L/h \end{bmatrix} \begin{bmatrix} V_{1,n} \\ V_{2,n} \\ I_{L,n} \end{bmatrix} + \begin{bmatrix} J_{n+1} \\ 0 \\ 0 \end{bmatrix}$$

Using the same steps as above, we can derive an expression for the trapezoidal formula (2):

$$\left(\mathbf{G} + \frac{2}{h}\mathbf{C}\right)\mathbf{X}_{n+1} = -\left(\mathbf{G} - \frac{2}{h}\mathbf{C}\right)\mathbf{X}_n + \mathbf{W}_{n+1} + \mathbf{W}_n \qquad (44)$$

NONLINEAR NETWORKS

Nonlinearities introduce major difficulties. The concepts of frequency-domain response, amplitude, phase, poles, or zeros do not exist. What remains is (1) a dc solution when no signal is applied and (2) a time-domain solution. Both are obtained by iterations.

Dc solutions find the operating point, which are nodal voltages and currents in the absence of a signal. It is a situation to which the network stabilizes after the power is turned on and no signal is applied. The operating point is found by first short-circuiting all inductors and open-circuiting (removing) all capacitors and then solving the resulting algebraic system. This is not without problems. In transistor networks we may have nodes connected to the rest of the network through capacitors only. Removal of capacitors will result in a node without connection to the other parts of the network, and in such a case the solution routines fail. Some kind of preprocessing may be needed to remove such nodes from the equations. Once this has been done, we have an algebraic system of equations which can be solved by Newton–Raphson iteration (see the section entitled "Solutions of Network Equations").

In linear networks we were able to write first the modified nodal equations and then put them into matrix form. This is not possible when nonlinear elements are present. Consider a nonlinear element connected between points i and j. Its current, which we denote as I_b, flows from i to j and is the function of the voltage across it, V_b:

$$I_b = F(V_b)$$

In terms of modified nodal formulation the branch voltage is expressed by

$$V_b = V_i - V_j$$

In nodal equations the current will be added at node i and subtracted at j,

$$\begin{aligned} &\textit{Equation}\ \text{for}\ \textit{node}\ i: \quad \ldots + F(V_i - V_j) \\ &\textit{Equation}\ \text{for}\ \textit{node}\ j: \quad \ldots - F(V_i - V_j) \end{aligned}$$

The dots indicate possible presence of other elements. To prepare the Jacobian, we differentiate with respect to V_i and V_j. Using the chain rule we obtain

$$\frac{\partial F}{\partial V_i} = \frac{\partial F}{\partial V_b}\frac{\partial V_b}{\partial V_i} = +\frac{\partial F}{\partial V_b}$$
$$\frac{\partial F}{\partial V_j} = \frac{\partial F}{\partial V_b}\frac{\partial V_b}{\partial V_j} = -\frac{\partial F}{\partial V_b}$$

Contribution to the Jacobian will be in columns and rows i and j:

$$\textit{Jacobian}: \quad \begin{bmatrix} \ldots + \partial F/\partial V_b & \ldots - \partial F/\partial V_b \\ \ldots - \partial F/\partial V_b & \ldots + \partial F/\partial V_b \end{bmatrix}$$
$$\textit{Right-hand side}: \quad \begin{bmatrix} \ldots + F(V_b) \\ \ldots - F(V_b) \end{bmatrix}$$

For additional understanding consider Fig. 17 with two linear and one nonlinear conductor. Using nodal formulation we take the sums of currents at each node:

$$f_1 = G_1V_1 + I_b - J = 0$$
$$f_2 = -I_b + G_2V_2 = 0$$

The expressions are already in the form needed for iteration, with zero on the right, see Eq. (18). The Newton–Raphson equation will have the form

$$\begin{bmatrix} G_1 + \partial F/\partial V_b & -\partial F/\partial V_b \\ -\partial F/\partial V_b & G_2 + \partial F/\partial V_b \end{bmatrix} \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \end{bmatrix} = -\begin{bmatrix} G_1V_1 + F(V_b) - J \\ -F(V_b) + G_2V_2 \end{bmatrix}$$

The minus sign in front of the right-hand side comes from Eq. (22). Had we used a linear conductance G_3 instead of the nonlinearity, it would appear in the same positions as the derivatives. We are coming to a very important conclusion: The derivative appears in the Jacobian in the same position as if the element was linear. All the stamps we derived for linear elements are also valid for the Jacobian.

Returning to Fig. 17, let $I_b = V_b^3$, $J = 1$, and $G_1 = G_2 = 1$. Then $\partial I_b/\partial V_b = 3V_b^2$ with $V_b = V_1 - V_2$. The Newton–Raphson

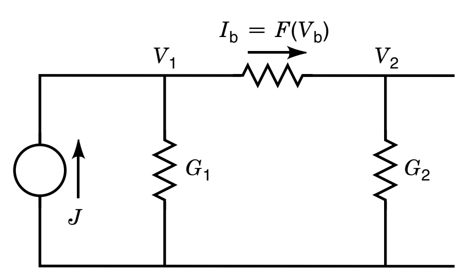

Figure 17. Resistive network with one nonlinear element.

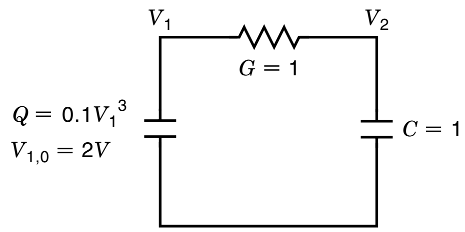

Figure 18. Network with nonlinear capacitor.

equation will be

$$\begin{bmatrix} 1+3(V_1-V_2)^2 & -3(V_1-V_2)^2 \\ -3(V_1-V_2)^2 & 1+3(V_1-V_2)^2 \end{bmatrix} \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \end{bmatrix} = -\begin{bmatrix} V_1+(V_1-V_2)^3-J \\ V_2-(V_1-V_2)^3 \end{bmatrix}$$

For iteration we must select some initial estimates on the voltages—for instance, $V_1 = 1$ and $V_2 = 0.5$.

Nonlinear memory elements *must* be defined by the flux, $\Phi(I_b)$, for the inductor and by the charge, $Q(V_b)$, for the capacitor. Simulation uses their derivatives with respect to time:

$$\begin{aligned} I_C &= \frac{\partial Q(V_b)}{\partial t} \\ V_L &= \frac{\partial \Phi(I_b)}{\partial t} \end{aligned} \tag{45}$$

and these derivatives are replaced by their approximations: for equal steps those from Table 1, for variable steps those from Table 2. Consider the network in Fig. 18 with a nonlinear capacitor defined by $Q = 0.1V_1^3$ and an initial voltage $V_{1,0} = 2V$. Writing nodal equations for both nodes

$$\begin{aligned} f_1 &= \frac{\partial Q}{\partial t} + G(V_1-V_2) = 0 \\ f_2 &= -G(V_1-V_2) + C\frac{\partial V_2}{\partial t} = 0 \end{aligned}$$

Suppose we use the backward Euler formula. This changes the equations to

$$\begin{aligned} f_1 &= \frac{0.1V_1^3 - 0.1V_{1,0}^3}{h} + G(V_1-V_2) = 0 \\ f_2 &= -G(V_1-V_2) + C\frac{V_2 - V_{2,0}}{h} = 0 \end{aligned} \tag{46}$$

The Jacobian is prepared by differentiating with respect to V_1 and V_2. This will lead to the following Newton–Raphson equation:

$$\begin{bmatrix} G+0.3V_1^2/h & -G \\ -G & G+C/h \end{bmatrix} \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \end{bmatrix} = \begin{bmatrix} -f_1 \\ -f_2 \end{bmatrix}$$

where on the right we insert Eq. (46). Had we used higher-order integration, then in the Jacobian the terms divided by h would be multiplied by the corrector coefficient a_0 (see Table 2). Additional details can be found in Refs. 2, 15, and 16.

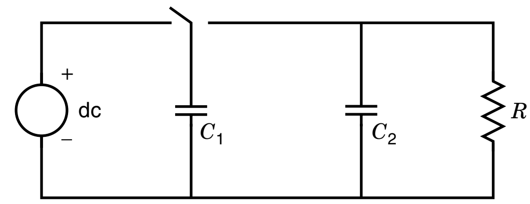

Figure 19. Network with Dirac impulse of current.

SWITCHED NETWORKS

In modern electronics, semiconductor devices can be used as reliable switches of voltage or current. This had led to the development of many methods where switched networks are used. In communications switched capacitor networks have been used successfully over the past two decades. In power engineering, classical power supplies were replaced by switched networks.

Simulation of networks with switches presents new challenges, not known before. Switching may be performed by a clock with switching instants known precisely in advance. In power engineering, diodes or transistors may be used as switches. Their switching instants depend on the system voltages or currents and change with time.

Detailed analysis of switching networks by classical simulators is difficult. Not only does the topology change, but the instants of switching change as well and have to be found with sufficient precision.

Modeling of switches can take various forms. Exact semiconductor models can be used, but then simulations are lengthy. The other possibility is to replace switches by open and short circuits, but this also presents problems. In this section we will explain what must be done if ideal switches are used.

Consider the network in Fig. 19, with the switch connected to the source. The capacitor C_1 is charged to the voltage of the voltage source, say V_1. The other node has a voltage $V_2 < V_1$. If we transfer the switch to the right, then we have a situation that at the same node is voltage V_1 from the left capacitor and V_2 from the right capacitor—a situation of inconsistent initial conditions. The voltages are equalized instantaneously by a Dirac impulse of current. A Dirac impulse is a strange impulse having zero duration and infinite amplitude, but finite area. Another case of inconsistent initial conditions is in Fig. 20. Let the switch be on the left side for some time. Since we have a dc source, a linearly growing current will flow upwards through the inductor. If we suddenly transfer the switch to the right, we will have a single loop with some current in the left inductor and zero current in the right induc-

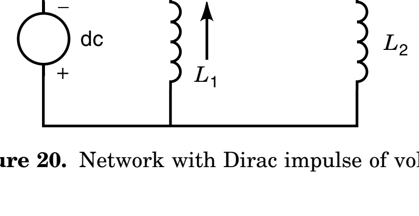

Figure 20. Network with Dirac impulse of voltage.

tor. The currents are instantaneously equalized by a Dirac impulse of voltage.

In network simulations the question is: What will be the voltages (in Fig. 19) or the currents (in Fig. 20) immediately after switching? The solution of the problem turns out to be very simple (17). All we have to do is use backward Euler integration formula, make a one-step h forward, and get the solution. Next use it as initial conditions for a step back, with negative h, to the instant of switching. The solution will provide the correct initial conditions after switching. Afterwards, integration is done as described in the previous section.

PERIODIC STEADY STATE

If a network is turned on from a quiescent state, there is always a certain period of time when transients take place. The same is true if we suddenly apply a signal. If the signal is a simple sine wave and the network is linear, the transients will eventually die out and the output will be the same sine wave, amplified or attenuated and with a different phase shift. In linear networks, this type of steady state is easily calculated using frequency domain methods.

If the network has nonlinearities and the signal is still a simple sine wave, the output will be composed of some transients, of the signal and its harmonics. In steady state the signal will be distorted. Because of the nonlinearity, classical frequency domain methods cannot be applied.

Networks with periodic steady state are quite common and designers need to know the behavior in steady state. Computer solution seems to be easy: use a periodic input signal and integrate for a sufficiently long time until all the transients have died out. Unfortunately, this may be a very expensive proposition, and it is thus no surprise that attempts have been made to somehow speed up the process to reach the steady state by other means.

Two fundamental types of methods are available: one is based on integration, the other on the use of frequency domain methods. We will explain the principles of both methods without going into any details. References will direct the reader to additional information.

To start integration, we need an initial vector of voltages and currents, $\mathbf{x}(0)$. If nothing more is known, we can start with a zero vector. Let us integrate over the period T of the periodic input signal and get the solution $\mathbf{x}(T)$. At this point we can form an error vector

$$\boldsymbol{E}(\boldsymbol{x}) = \boldsymbol{x}(T) - \boldsymbol{x}(0)$$

which, at steady state, must be a zero vector. To solve the problem by Newton–Raphson iteration, we would have to calculate the Jacobian, which is a fairly complicated process. Moreover, the first solution may not result in a zero vector and the process may have to be repeated. More about this method can be found in Refs. 18 and 19.

Another method, based on integration and not requiring the knowledge of the derivatives, was invented by Skelboe (20,21). The principle is as follows: integrate over a number of periods, save the vectors $\mathbf{x}(0)$, $\mathbf{x}(T)$, . . . $\mathbf{x}(nT)$, and apply a special ϵ algorithm to project the result to the steady state. Similarly as above, the projection may not be satisfactory in the first run, and the process may have to be repeated several times with new integrations over n periods.

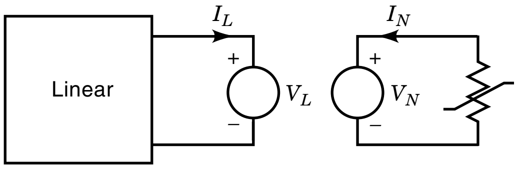

Figure 21. Linear network with one nonlinear resistor.

Methods based on frequency domain solutions completely avoid calculation of the transients; they are usually referred to as *harmonic balance* methods. We will explain the principle with the help of Fig. 21, composed of some linear network and one nonlinear resistor. Normally the linear and nonlinear part would be connected. In the figure we separated them and applied two sources $V_N = V_L$. We also indicated the currents I_N and I_L. If these two equal voltage sources are such that the two currents $I_L = I_N$, then we have a situation in which a direct connection of the nonlinearity to the linear network will not result in any change.

The trick of separating the two parts by the voltage sources has the advantage that we can apply harmonic frequencies f, $2f$, . . . nf to the linear network one by one (superposition principle applies) and get the currents by frequency domain methods. This will provide a vector of currents

$$\boldsymbol{I}_L = [I_{L,1}, I_{L,2}, \ldots I_{L,n}]^T$$

Next, we apply the same voltages to the nonlinear part, but first convert them to time domain by using Fast Fourier Transform (FFT). This will give us the possibility to calculate the current through the nonlinearity as a time-domain function. Using the inverse FFT we decompose this current into frequency-domain components:

$$\boldsymbol{I}_N = [I_{N,1}, I_{N,2}, \ldots I_{N,n}]^T$$

We can now create an error vector

$$\boldsymbol{E} = \boldsymbol{I}_L - \boldsymbol{I}_N$$

and using some iterative process try to reduce this vector to a zero vector. Additional details can be found in Ref. 22.

Many modifications of the above methods have been published. For further study we recommend Ref. 23. It is a book devoted to the steady-state problem and has numerous additional references on this subject. In addition, Refs. 24 and 25 may be of interest; they are books dealing with the general problem of circuit simulation.

BIBLIOGRAPHY

1. J. Vlach, *Basic Network Theory with Computer Applications,* New York: Van Nostrand Reinhold, 1992.
2. J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design,* 2nd ed., New York: Van Nostrand Reinhold, 1994.
3. G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computations,* Englewood Cliffs, NJ: Prentice-Hall, 1977.
4. G. Dahlquist and A. Bjorck, *Numerical Methods,* Englewood Cliffs, NJ: Prentice-Hall, 1974.
5. D. Kahaner, C. Moler, and S. Nash, *Numerical Methods and Software,* Englewood Cliffs, NJ: Prentice-Hall, 1989.

6. R. A. Rohrer, *Circuit Theory: An Introduction to the State Variable Approach,* New York: McGraw-Hill, 1970.
7. P. M. DeRusso, R. J. Roy, and C. M. Close, *State Variables for Engineers,* New York: Wiley, 1965.
8. G. D. Hachtel, R. K. Brayton, and F. G. Gustavson, The sparse tableau approach to network analysis and design, *IEEE Trans. Circuits Syst.,* **CT-18**: 101–113, 1971.
9. C. W. Ho, A. E. Ruehli, and P. A. Brennan, The modified nodal approach to network analysis, *IEEE Trans. Circuits Syst.,* **CAS-22**: 504–509, 1975.
10. G. Dahlquist, A special stability problem for linear multistep methods, *BIT,* **3**: 27–43, 1963.
11. C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations,* Englewood Cliffs, NJ: Prentice-Hall, 1971.
12. R. K. Brayton, Error estimates for the variable-step backward differentiation methods, IBM Research Report RC 6205, September 1976.
13. R. K. Brayton, F. G. Gustavson, and G. D. Hachtel, A new efficient algorithm for solving differential-algebraic systems using implicit backward differentiation formulas, *Proc. IEEE,* **60**: 98–108, 1972.
14. I. N. Hajj and S. Skelboe, Time domain analysis of nonlinear systems with finite number of continuous derivatives, *IEEE Trans. Circuits Syst.,* **CAS-26**: 297–303, 1979.
15. C. W. Gear, Simultaneous numerical solution of differential-algebraic equations, *IEEE Trans. Circuit Theory,* **CT-18**: 89–95, 1971.
16. W. M. G. Van Bokhoven, Linear implicit differentiation formulas of variable step and order, *IEEE Trans. Circuits Syst.* **CAS-22**: 109–113, 1975.
17. J. Vlach, J. Wojciechowski, and A. Opal, Analysis of nonlinear networks with inconsistent initial conditions, *IEEE Trans. Circuits Syst. I,* **42**: 1995, 195–200.
18. T. J. Aprille and T. N. Trick, Steady state analysis of nonlinear circuits with periodic inputs, *Proc. IEEE,* **60** (1): 108–114, 1972.
19. F. R. Colon and T. N. Trick, Fast periodic steady state analysis for large signal electronic circuits, *IEEE J. Solid State Circ.,* **8** (4): 260–269, 1973.
20. S. Skelboe, Computation of the periodic steady state response of nonlinear networks by extrapolation methods, *IEEE Trans. Circuits Syst.,* **CAS-27**: 161–175, 1980.
21. S. Skelboe, Conditions for quadratic convergence of quick periodic steady state methods, *IEEE Trans. Circuits Syst.,* **CAS-28** (4): 234–239, 1982.
22. M. S. Nakhla and J. Vlach, A piecewise harmonic balance technique for determination of periodic response of nonlinear systems, *IEEE Trans. Circuits Syst.,* **CAS-23** (2): 85–91, 1976.
23. K. S. Kundert, J. K. White, and A. Sangiovanni-Vincentelli, *Steady-State Methods for Simulating Analog and Microwave Circuits,* Norwell, MA: Kluwer, 1990.
24. J. Ogrodzki, *Circuit Simulation Methods and Algorithms,* Boca Raton, FL: CRC Press, 1994.
25. L. T. Pillage, R. A. Rohrer, and Ch. Visweswariah, *Electronic Circuit and System Simulation Methods,* New York: McGraw-Hill, 1995.

JIRI VLACH
University of Waterloo

TIME-DOMAIN REFLECTOMETERS. See REFLECTOMETERS, TIME-DOMAIN.

TIME-FREQUENCY ANALYSIS. See FREQUENCY MODULATION.

TIME HOPPING MODULATION. See SPREAD SPECTRUM COMMUNICATION.

TIME INTERVAL METERS. See FREQUENCY AND TIME INTERVAL METERS.