PIECEWISE-LINEAR TECHNIQUES

Simulation programs play an important role in the design of integrated electronic systems. They allow the designer to collect information on the performance of the system that is being designed before that the system is actually realized. To do so, the circuit is described as a collection of separate modules that are connected in some way. Depending on the type of circuit, these modules are of a different nature (e.g., transistors, logic gates, behavioral models), each with their own corresponding data structure and typical solution algorithm. Within a certain application, modules of different complexity can also be used to supply variable detail in the resolution of the circuit response that must be calculated. For fast and efficient simulation, the algorithms to solve the set of equations describing the modules' behaviors are highly optimized with respect to storage requirements, accuracy, or convergence speed. As a result, it is nearly impossible to combine the analysis for all different aspects in one single run using conventional analysis methods.

Often an approach is followed to construct and apply some artificial interfaces between the different types of modules that allow for separate analysis of each subsystem but also serve as an interconnection for exchanging the response data between the modules. Well known practical solutions are simulation back planes or close coupling of a circuit level simulator with a digital simulator. In general, these methods suffer from large storage requirements, diverging iterations, and slow computational speed. Furthermore, they lack the necessary flexibility to be applied to a broad class of problems.

The creation of a mathematical description that approximates the system's functionality is called *modeling* and the description itself the *model description* or simply the *model.* The aforementioned problems can be avoided to a large extent when a common model is used that has a single solution algorithm to solve the overall system response. The mathematical description of such model has to be flexible enough to cover the input-output description of a broad class of modules (e.g., device modules at the voltage-current level, logic modules at the Boolean level, behavioral modules). A model description that could deal with a large range of nonlinear multidimensional functions is suitable for that purpose when it permits the formulation of the relevant equations and its linked to a suitable solution strategy. The standard approach in circuit level simulation is to use analytical functions for the relations, and the key algorithm for the solution process is the well-known Newton–Raphson (NR) iteration. This method generates a sequence of iterates that (it is hoped) converges to the required solution using derivatives of the modules' equations. The limitations of the use of an NR scheme are various, and the important ones are the local behavior of the method (not all solutions can be obtained); a sufficient close guess for the solution, which is required as a starting point; and the computational burden of a repetitive inversion of the derivatives.

Many of the aforementioned problems can be prevented or solved using a different type of modeling, the so-called *piecewise linear* (PL) modeling. Here the nonlinear behavior of the modules' analytical expression is replaced by a collection of linear relations in a sequence of adjacent intervals. The immediate advantage of a piecewise linear approach is that the local relation between the variables is always linear except at the boundaries, which may simplify further computations. The close mathematical relation of PL modeling and linear algebra can be beneficial in nonlinear network theory. An obvious drawback is the limited precision obtained, which can only be avoided by increasing the number of linear relations to approximate the nonlinear behavior at the cost of a higher computational load. The use of PL modeling results in a special data structure that makes it possible to use solution algorithms with a global convergence behavior. In the case of computing the direct current (dc) operating point of a circuit, this results in less restrictions on the initial guess of the starting point. Furthermore, this property is advantageous when a circuit with many dc operating points is to be analyzed. It is due to these properties that piecewise linear techniques are used today in modern simulators to find dc operating points.

PIECEWISE LINEAR MODEL DESCRIPTIONS

Confronted with the question to develop a piecewise linear model for nonlinear components in electrical circuits, one obviously starts to look for the most simple extension to the well-known linear components like resistors and linearly controlled sources. This extension should in one way supply us with a kind of basic nonlinearity, but in another way this nonlinearity should be as simple as possible, with the expectation of extending this approach to more general nonlinearities later on. The first component that will come up to satisfy those conditions seems to be the semiconductor diode. It surely is one of the most simple nonlinear elements and has been used for a long time already to synthesize or reproduce nonlinear transfer functions in analog computers by realizing piecewise continuous approximations. One can try to idealize the behavior of a diode. An ideal diode draws no reverse current when polarized into reverse bias and does not need any forward bias voltage to conduct an arbitrary forward current. Such an idealization yields a v-i relation that consists of only two branches, one described by $v = 0$ and $i > 0$ and one by $v < 0$ and $i = 0$. For reasons of symmetry, we will reverse the voltage reference polarity of the ideal diode with respect to the normal convention such that the characteristic now reads

$$v, i \geq 0 \text{ and } v \cdot i = 0 \tag{1}$$

Figure 1 shows the relation between the characteristics of an actual diode, an ideal electrical diode, and the ideal diode as used in the context of PL. Note that the characteristic of the ideal diode can also be considered as being piecewise linear by itself, with 2 being the minimum number of PL segments necessary to differentiate the diode from fully linear elements. In this respect this diode indeed seems very basic.

In any actual electrical network application, this element can only exist in one of two possible states—it either conducts with zero voltage, representing a closed connection between its terminals, or it blocks the current in the reverse mode, behaving as an open circuit. This means that any linear circuit containing ideal PL diodes only changes its topology when these diodes switch from the conducting state (i.e., switch from short to open circuit, or the other way around). Therefore, the response of the network will remain linear in any conducting state of the diodes, but for different conduction states the response will be different since we deal with a network with switches that can change the topology. As the switching occurs in the point $v = i = 0$, the response will automatically be continuous for the applied excitations. This property is essential and will be used to advantage in the context of the finding of all dc operating points of networks.

Explicit Piecewise Linear Models

Figure 2 shows a fairly simple network in which a number of resistors, independent voltage sources (batteries), and diodes are connected in parallel. In each parallel branch, the ideal diode starts to conduct when the input voltage exceeds the voltage of the battery. The consecutive independent sources increase in voltage (i.e., $e_1 < e_2$). This means that an increasingly higher voltage is required as the input to include the parallel branches that are placed more to the right in the figure. However, in case these branches start to conduct, the total resistance is decreasing (increasing) when the resistor's value is positive (negative) and hence the slope of the current-voltage characteristic is increasing (decreasing), leading to the v-i characteristic as also depicted in Fig. 2. This fairly simple network will hereafter be treated as a nonlinear resistor with a piecewise linear behavior.

If we are able to describe the electrical behavior of the network of Fig. 2, we will obtain a mathematical description of a one-dimensional PL function without any further restrictions. Should this network description result in an explicit solution, this would yield an explicit PL function. However, it will always produce at least an implicit description. From arguments from electrical network theory, we know that it is possible to construct a dual electrical network that has the same functional relation with the roles of current and voltage interchanged. Hence we immediately conclude that the description that we are looking for will not be unique.

In the preceding situation it is fairly easy to produce an explicit description of the v-i relation at the input terminals using basic mathematical functions. To this purpose consider the following expression:

$$\lfloor x \rfloor = \frac{1}{2}(x + |x|) \tag{2}$$

which realizes a ramp function with the breakpoint at $x = 0$. Based on our previous discussion, using Eq. (2) the current in branch k satisfies for $k > 0$

$$i_k = G_k \lfloor v - e_k \rfloor \text{ with } G_k = 1/R_k \tag{3}$$

Application of Eq. (3) and summation over all branches immediately yields

$$i = G_1(v - e_1) + \sum_{k=2}^{n} G_k \lfloor v - e_k \rfloor$$

or

$$i = G_1(v - e_1) + \frac{1}{2}\sum_{k=2}^{n} G_k(v - e_k) + \frac{1}{2}\sum_{k=2}^{n} G_k|v - e_k| \tag{4}$$

which, for the example of Fig. 2, leads to

$$i = v - \frac{3}{4} - \frac{3}{4}|v - 1| + \frac{3}{4}|v - 2| \tag{5}$$

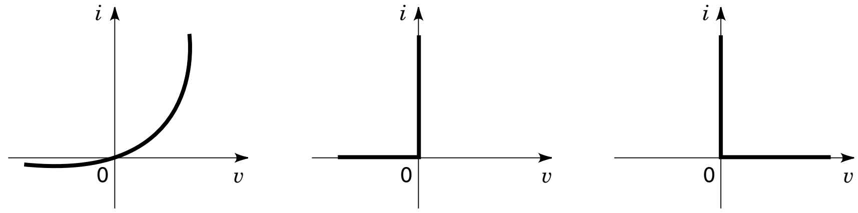

Figure 1. The transformation of an actual diode characteristic into the ideal electrical behavior and into the ideal behavior as defined in Eq. (1).

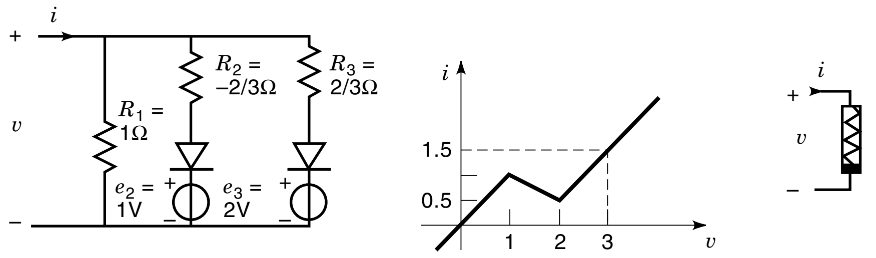

Figure 2. A circuit example with ideal diodes. The circuit can be represented as a nonlinear resistor with a piecewise linear behavior as defined by the v-i characteristic representing the behavior as seen from the port nodes.

In a more general mathematical expression, the model description for the PL function $\boldsymbol{f} : R^n \to R^m$ is given by

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{a} + B\boldsymbol{x} + \sum_{i=1}^{\sigma} \boldsymbol{c}_i |\langle \boldsymbol{\alpha}_i, \boldsymbol{x}\rangle - \beta_i| \tag{6}$$

where $B \in R^{m\times n}$, a, $c_i \in R^m$, $\alpha_i \in R^n$, and $\beta_i \in R^1$ for $i \in \{1, \ldots, \sigma\}$ and which is the basic model description as proposed by Chua and Kang (1–3). In the model description, hyperplane H_i is expressed as

$$\langle \boldsymbol{\alpha}_i, \boldsymbol{x}\rangle - \beta_i = \alpha_i^T x - \beta_i = 0 \tag{7}$$

This hyperplane H_i divides the domain space into two regions, R_{1i} and R_{2i}. The normal vector of the plane is defined by α_i. The hyperplane reflects the operation of the ideal diode i, one region corresponding to the situation in which this diode conducts and the other to its blocking state. This can also be seen from Eq. (4), in which each absolute-sign operator refers to an ideal diode in the network. Using the model definition, the domain space R^n is divided into a finite number of polyhedral regions by σ hyperplanes H_i of dimension $n - 1$. When crossing H_i, the Jacobian matrix J of Eq. (6) changes with the amount

$$\Delta J = J_{1i} - J_{2i} = (-\boldsymbol{c}_i \boldsymbol{\alpha}_i^T) - (\boldsymbol{c}_i \boldsymbol{\alpha}_i^T) = -2\boldsymbol{c}_i \boldsymbol{\alpha}_i^T \tag{8}$$

Notice that this amount is independent of the position in R^n where the hyperplane is crossed. This property is known as the *consistent variation* property and plays an important role in piecewise linear modeling (3).

Each one-dimensional function or any one-port electrical network with ideal diodes and linear elements can be realized by Eq. (6), and there exists a one-to-one relation between the parameters in Eq. (6) and the given piecewise linear function or network. Consider again the nonlinear resistor in Fig. 2, and notice that all elements of the network describing the nonlinear resistor are used exactly once in the model Eq. (5). However, in more dimensions hyperplanes can cross each other, and geometrical constraints might exist, such that not all multidimensional functions can be represented by this model description. In terms of an electrical network this means that not only linear components are used but, for instance, also controlled sources.

Therefore, people have tried to extend this model description to allow modeling of higher-dimensional piecewise linear functions. Assume that Eq. (2) can be considered as a base operation of order one, given as

$$u_1 = \lfloor f_1(x) \rfloor$$

and that the second-order base function looks like

$$u_2 = \lfloor f_2(x) + a_{21} \lfloor f_1(x) \rfloor \rfloor$$

Then it can be proven that using this extension any two-dimensional function or two-port electrical network can be modeled (4,5). Here we assume that the functions $f_i(x)$, $i = 1, 2$ are affine functions. Figure 3 shows a geometrical interpretation of these base functions. In a two-dimensional situation, hyperplanes may cross each other and a hyperplane itself may eventually be piecewise linear under the condition that the breakpoint is defined by a hyperplane described by a base function of order one. In a similar way, we can define base function i

$$u_i = \lfloor f_i(x) + \sum_{k=1}^{i-1} a_{ik} u_k \rfloor \tag{9}$$

and with this set of base functions it can be proven that any PL function or any multiport can be modeled (6,7). However, the function or network should be of class P, a property we will discuss later.

Implicit Piecewise Linear Models

We can consider the circuit in Fig. 2 as a special case of a linear memoryless electrical multiport network that is loaded at some of its ports by the previously defined ideal diodes. The network may contain resistors and fixed and controlled sources and, for later convenience, all of its ports are partitioned in two different sets, port set 1 and port set 2. Figure

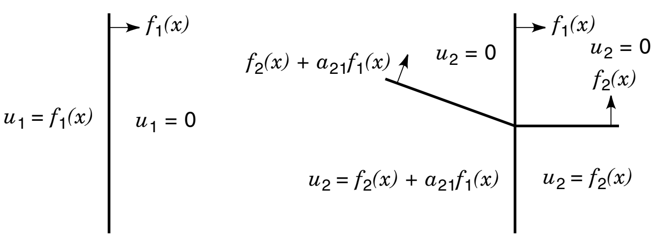

Figure 3. The first- and second-order base functions with respect to their hyperplane(s). In the case of second-order base functions, hyperplanes may be piecewise linear. In each half-space, the function's value is given and for each hyperplane the normal vector is given also.

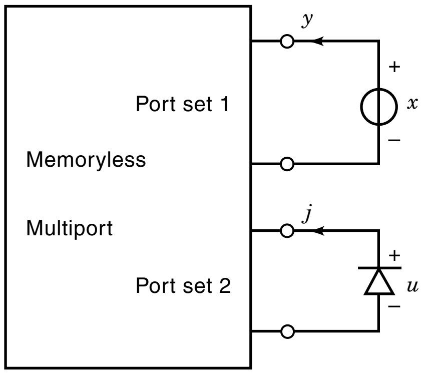

Figure 4. A memoryless electrical multiport loaded at port set 2 with ideal diodes. The voltage (current) across (through) each diode is represented by $u(j)$. Port set 1 represents the independent variables x and the dependent variables y.

4 shows such a network. Assume that port set 1 contains m ports and port set 2 contains k ports. For port set 2 all its ports are loaded by ideal diodes; thus the voltage u over the diodes and the current j through the diodes can be represented by k-dimensional vectors (i.e., $\boldsymbol{u}, \boldsymbol{j} \in R^k$). Then, based on the previous discussion, the conducting states of the diodes will depend on the m currents and m voltages at port set 1. Next assume (without loss of generality) that the port set 1 is excited by voltage sources. In the light of the PL model that we are required to construct, we map the port variables at port set 1 on new vectors $\boldsymbol{x}, \boldsymbol{y} \in R^m$ according to $x = v$ and $y = i$.

Diode k has only two states that are separated by the condition $u_k j_k = 0$. The diode now also separates the space spanned by x and y into two half-spaces, one in which the diode conducts and one in which it blocks. The boundary between the two half-spaces is a hyperplane and, as stated before, is determined by $u_k j_k = 0$. Since in any conducting state the response of the network will remain linear in terms of the applied voltage excitation, the components of the vectors $\boldsymbol{y}$, $\boldsymbol{u}$, and $\boldsymbol{j}$ are all linear relations in the components x_i of the vector $\boldsymbol{x}$. Hence the hyperplane can be rewritten such that a linear combination of the components x_i of $\boldsymbol{x}$ are equal to zero. That is,

$$\boldsymbol{c}^t\boldsymbol{x} + g = 0 \tag{10}$$

All the diodes together separate the complete input space into 2^k polyhedral regions, called polytopes. Within each polytope, all diodes remain in one of their states; some will conduct and others will block. Within each polytope, we have a linear relation between x and y. Crossing a hyperplane means that the diode corresponding to this hyperplane changes its state and hence we have an other topology, defined by the polytope in which we enter after crossing the boundary. Again we are confronted with a linear network.

Consider the situation that we have k diodes or hyperplanes, and therefore 2^k polytopes. For each polytope, denoted by K_m, we have a linear mapping representing the topology of the network for that polytope:

$$\boldsymbol{y} = A_m\boldsymbol{x} + \boldsymbol{f}_m \qquad m = 1, 2, \ldots, 2^k \tag{11}$$

The complete set of Eq. (11) describes a PL mapping that is defined on a collection of polytopes. The boundaries of each polytope K_m will be formed by a set of bounding hyperplanes H^i_m according to

$$H^i_m = \{\boldsymbol{x} | \boldsymbol{c}^i_m\boldsymbol{x} + g^i_m = 0\}, i \in \{1, \ldots, k\} \tag{12}$$

which is a generalization of Eq. (10). Equation (12) defines a collection of half-spaces V^i_m given by

$$V^i_m = \{\boldsymbol{x} | \boldsymbol{c}^i_m\boldsymbol{x} + g^i_m \geq 0\} \text{ and } K_m = \bigcap_i V^i_m \tag{13}$$

The row vectors $\boldsymbol{c}^i_m$ in Eq. (13) are the normal vectors on the hyperplanes that bound the polytope K_m and point in the inward direction. They all can be considered as rows of a matrix C_m such that the polytope K_m is equivalently given by

$$K_m = \{\boldsymbol{x} | C_m\boldsymbol{x} + \boldsymbol{g}_m \geq 0\} \tag{14}$$

Define each polytope K_m by determining on which side of each hyperplane H^i it is situated. Note that for the hyperplanes the subscript m is removed to express that we only have a single set of hyperplanes H^i, $i \in \{1, \ldots, k\}$, which can partition the space into a maximum of 2^k polytopes. This exactly fits with our network with ideal diodes. The k diodes can define at most 2^k topologies into which the network can be divided. Therefore, for any polytope K_m and each hyperplane H^i we either have one of two possibilities:

$$C^i\boldsymbol{x} + \boldsymbol{g}^i \geq \boldsymbol{0} \quad \text{or} \quad C^i\boldsymbol{x} + \boldsymbol{g}^i < \boldsymbol{0} \tag{15}$$

Because the normal vectors of the hyperplanes were considered to be rows of C_m and thus also for C^i, we may collect all normal vectors into a single matrix C. The same holds for vector $\boldsymbol{g}$. Once a partitioning of the space is given, the various matrices A_m and vectors $\boldsymbol{f}_m$ in Eq. (11) also have to be defined in accordance with Eq. (15). We are, of course, looking for a compact description of the piecewise linear function as defined by the network with ideal diodes. From a network point of view it is clear that the network is continuous and hence the underlying piecewise linear function. As a result, the matrices and vectors in Eq. (11) become related and may not freely be chosen. This dependency is the same as expressed by Eq. (8), yielding in this situation (assuming separation hyperplane H^p)

$$A_i = A_j + \frac{(\boldsymbol{f}_i - \boldsymbol{f}_j)C_{p\cdot}}{g_p} \tag{16}$$

which fully determines the relation between two mappings from adjacent regions. Now the piecewise linear function is described completely by relation Eq. (11) together with the description of the state space. Again consider the network of Fig. 4, from which we learned that its response is a piecewise linear function that could be used to derive a closed form expression for a piecewise linear mapping. From the v-i curves of the ideal diodes as given in Eq. (1), we recall that for each diode at port set 2 we have

$$\boldsymbol{u}, \boldsymbol{j} \geq 0 \quad \boldsymbol{u}^T\boldsymbol{j} = 0 \tag{17}$$

with the inequalities taken component wise. Furthermore, we assume that the electrical behavior of the network within the solid box at its outside ports can be described by a port-admittance matrix H, resulting in

$$\begin{bmatrix} \boldsymbol{i}_1 \\ \boldsymbol{i}_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{bmatrix}$$

Renaming $\boldsymbol{i}_1$ and $\boldsymbol{v}_1$ into $\boldsymbol{y}$ and $\boldsymbol{x}$ and the variables of port set 2 into $\boldsymbol{u}$ and $\boldsymbol{j}$ (because they are related to the diodes) and substituting Eq. (16) yields

$$\boldsymbol{y} = A\boldsymbol{x} + B\boldsymbol{u} + \boldsymbol{f} \tag{18}$$

$$\boldsymbol{j} = C\boldsymbol{x} + D\boldsymbol{u} + \boldsymbol{g} \tag{19}$$

$$\boldsymbol{u}, \boldsymbol{j} \geq 0 \quad \boldsymbol{u}^T\boldsymbol{j} = 0 \tag{20}$$

which is known as the *state model* of a PL mapping $\boldsymbol{f} : \boldsymbol{x} \rightarrow \boldsymbol{y}$ (8). Equation (18) determines the input-output mapping of $\boldsymbol{x}$ onto $\boldsymbol{y}$. The remaining two equations determine the state of the mapping from the electrical state of the ideal diodes. These diodes form a kind of state variables, which, together with the input vector $\boldsymbol{x}$ determine the output y comparable to the situation in a state-space model of a linear dynamic system. The conditions in Eq. (20) are called the *complementary conditions* and $\boldsymbol{u}$ and $\boldsymbol{j}$ are complementary vectors. It is obvious that some algebraic mechanism will be needed to be able to use the PL mapping in an efficient way. Storage and updating of the description of the mappings as well as the calculation of the mapping itself can then be performed by standard operations from linear algebra.

A few years after the publication of this model description, a new model was introduced in which the hyperplanes were allowed to be situated in the image space. However, the matrix in front of the state vector $\boldsymbol{u}$ in the state equation should then be the identity matrix, resulting in the description

$$I\boldsymbol{y} + A\boldsymbol{x} + B\boldsymbol{u} + \boldsymbol{f} = 0 \tag{21}$$

$$\boldsymbol{j} = C\boldsymbol{x} + D\boldsymbol{y} + I\boldsymbol{u} + \boldsymbol{g} \tag{22}$$

where Eq. (20) still holds (9).

By now it should be clear that any piecewise linear memoryless electrical multiport can be described by Eqs. (18) to (20). However, many networks can be handled by the description of Eqs. (21) and (22), which has some advantages with respect to analysis. To allow efficient analysis, it is important that after a diode changes its conductivity and hence the topology of the network is changed, the new description of the network can be obtained efficiently (10). Because the state matrix in front of the state vector $\boldsymbol{u}$ in Eq. (22) is the identity matrix, only Eq. (21) has to be modified during a topology change of the network. As modeling example, consider the model description of Eqs. (18) to (20) for the nonlinear resistor in Fig. 2, which can be written as

$$\begin{aligned} & i + (-1)v + \left(\frac{3}{2} \; -\frac{3}{2}\right)\boldsymbol{u} = 0 \\ & j = \begin{pmatrix} -1 \\ -1 \end{pmatrix} v + I\boldsymbol{u} + \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\ & \boldsymbol{u}, \boldsymbol{j} \geq 0 \quad \boldsymbol{u}^T\boldsymbol{j} = 0 \end{aligned} \tag{23}$$

and which is a format similar to Eqs. (21) and (22). We can easily show that for any one-dimensional one-to-one function this property holds.

Relations Between Piecewise Linear Model Descriptions

To compare explicit and implicit model description in order to rank them, let us define the modulus operator:

Definition 1. Let $\boldsymbol{z}, \boldsymbol{u}, \boldsymbol{j} \in R^n$ and let the n-dimensional vector function $\boldsymbol{\phi}(\cdot)$ be given as $\boldsymbol{\phi}(\boldsymbol{z})_k = h(z_k)$, where the subscript k denotes the kth element of a vector and $h(\cdot)$ is a scalar function. For a strictly increasing h: $R_+ \rightarrow R_+$ and $h(0) = 0$, the transformation $\boldsymbol{z} \rightarrow \boldsymbol{u}, \boldsymbol{j}$ defined by $\boldsymbol{u} = \boldsymbol{\phi}(|\boldsymbol{z}| + \boldsymbol{z}), \boldsymbol{j} = \boldsymbol{\phi}(|\boldsymbol{z}| - \boldsymbol{z})$ is called the modulus transformation.

The modulus transformation automatically guarantees that $\boldsymbol{u} \geq 0, \boldsymbol{j} \geq 0, \boldsymbol{u}^T\boldsymbol{j} = 0$, which exactly matches Eq. (20). If we define $h(\cdot)$ as $h(t) = t$. Corollary 1 immediately follows from Definition 1:

Corollary 1. The modulus transform for $h(t) = t$ is equivalent to the mapping $u, j \rightarrow z$ satisfying $|z| = (u + j)/2$ and $z = (u - j)/2$, with $z \in R$ and $u, j \in R_+$.

By this corollary we have an operator to compare the implicit model description, which uses complementary vectors, with the explicit model description, as described by absolute-sign operators. Each explicit model description can be rewritten into a format similar to Eqs. (18) to (20) (10). To compare the descriptions, we only have to compare the obtained matrix in front of the state vector $\boldsymbol{u}$. Doing so leads to the conclusion that all explicit model descriptions are a subclass of the description Eqs. (18) to (20). If we have a description using base functions Eq. (9), it covers at maximum any PL function for which the matrix D in Eq. (19) is of class P:

Definition 2. A matrix D belongs to class P if and only if $\forall z \in R^p, z \neq 0, \exists k{:}\ z_k \cdot (Dz)_k > 0$.

Class P is alternatively defined by the property that all principal minors of D are positive. The model description of Eqs. (21) and (22) is also a subclass of Eqs. (18) to (20) but covers a larger class than Eq. (6). Note that the description of Eqs. (18) to (20) also allows modeling of functions not being of class P. As an example, consider the one-to-many mapping

$$\begin{aligned} & x \leq 1 \,\} \quad f(x) = 0 \\ & \left.\begin{matrix} x \leq 1 \\ x \geq 0 \end{matrix}\right\} \quad f(x) = -x + 1 \\ & x \geq 0 \,\} \quad f(x) = 1 \end{aligned}$$

for which the description yields

$$\begin{aligned} & y = (-1)x + (-1 \quad 1)\boldsymbol{u} + (1) \\ & \begin{pmatrix} j_1 \\ j_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} x + \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \boldsymbol{u} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{aligned}$$

having a matrix D not of class P. Such a function cannot be described by any explicit model description.

Although each model format does not change with respect to the function or network to be modeled, the model size is strongly related to the number of ideal diodes in the network or linear segments in the function. The more linear descriptions are used to approximate the nonlinear behavior, the larger the data storage will be, and this relation yields a linear behavior. However, the complexity to solve the model to obtain an output for a given input later on increases exponentially with the number of ideal diodes in the network. In the explicit models this can be seen from the evaluation of the absolute-sign operators and for the implicit models from the evaluation of the complementary conditions. In both situations we have to check the two sides of each diode that is added to the network.

For many practical situations, piecewise linear models for the electrical elements can be obtained easily (11). This holds for a device element described at the current-voltage level, but also for digital components in terms of Boolean algebra or behavioral models of, for example, complete analog to digital (AD) or digital to analog (DA) converters. Also, time-dependent elements such as capacitors or even differential equations can be described when we modify the implicit model descriptions (9,11). Although most PL models are generated by hand, automatic model generators do exist for several functions (12,13).

SOLUTION ALGORITHMS

When we are using explicit model descriptions, we only have to solve the absolute-sign operators, which is an evaluation task. However, in the case of an implicit model, which is more powerful and is therefore more used in circuit modeling, we have to obtain the internal state variables by solving the state equations. Without any restrictions, we assume that the electrical network is described in terms of Eqs. (21) and (22) and that we know that the description is valid for $\boldsymbol{u} = \boldsymbol{0}$. Then we may use the linear mapping to eliminate the output vector in the state equation, yielding

$$\boldsymbol{j} = (C - DA)\boldsymbol{x} + I\boldsymbol{u} + (\boldsymbol{g} - D\boldsymbol{f}) \tag{24}$$

which can be transformed into

$$\boldsymbol{j} = I\boldsymbol{u} + \boldsymbol{q} \tag{25}$$

where $\boldsymbol{q} = (C - DA)\boldsymbol{x} + (\boldsymbol{g} - D\boldsymbol{f})$. This equation is a special case of

$$\begin{aligned} &\boldsymbol{j} = M\boldsymbol{u} + \boldsymbol{q} \\ &\boldsymbol{u} \geq 0, \boldsymbol{j} \geq 0, \boldsymbol{u}^T\boldsymbol{j} = 0 \end{aligned} \tag{26}$$

where for a given $\boldsymbol{q}$ the complementary vectors $\boldsymbol{u}$ and $\boldsymbol{j}$ should be solved.

The problem defined by Eq. (26) is known as the linear complementary problem (LCP) and the solution to this problem is the key operation in the evaluation of a PL function based on Eqs. (18) to (20) or Eqs. (21) and (22). The LCP has been known as a basic problem for quite some time and is mainly studied for applications in game theory and economics (14,15). In the past 20 years a number of algorithms have been developed to solve the LCP, which in its most general form is known to be an NP-complete problem. The solution can be found by going through all possible so-called pivotisations of matrix M, which number is exponential in the dimension of M. A more efficient approach is to construct algorithms that use an extension of a local solution estimate to find the required result. Note that the dimension of M depends on the number of linear segments used to approximate the nonlinear behavior of a function.

The most well-known method for this purpose is the homotopy algorithm by Katzenelson (16). Katzenelson introduced this method in 1965, and the method is still extensively used in piecewise linear simulation programs (10,17,18). Being a homotopy method, a continuous path through the space is created by extending the LCP of Eq. (26) according to

$$\boldsymbol{j} = M\boldsymbol{u} + \boldsymbol{q}_0 + \lambda(\boldsymbol{q}^* - \boldsymbol{q}_0) \tag{27}$$

where we assume that $\boldsymbol{q}_0$ is known with $\boldsymbol{u} = 0, \boldsymbol{j} > 0$ for λ is zero. We are looking for the solution for a $\boldsymbol{q}^*$. The homotopy parameter λ is to be increased from zero to one. The procedure is to gradually increase parameter λ until a component j_m becomes zero, because $[\boldsymbol{q}_0 + \lambda(\boldsymbol{q}^* - \boldsymbol{q}_0)]|_m = q_m$ becomes zero. Just a small increase of λ is needed to let $u_m > 0$, thus preventing j_m from becoming negative, which is not allowed according to the complementary conditions. In terms of the electrical network, this means that diode m is changing its state and the network topology is changing. We have to perform a pivoting operation with again a system of equations according to Eq. (27). The pivot is the diagonal element M_{mm}, which we assume to be positive. As a result, variables j_m and u_m will change place and Eq. (27) changes into a new form given by

$$\boldsymbol{v} = \overline{M}\boldsymbol{w} + \overline{\boldsymbol{q}}_0 + \lambda(\overline{\boldsymbol{q}}^* - \overline{\boldsymbol{q}}_0), \qquad \boldsymbol{v}, \boldsymbol{w} \geq 0, \boldsymbol{v}^T\boldsymbol{w} = 0 \tag{28}$$

in which $\boldsymbol{w} = 0$ and $\boldsymbol{v} = \overline{\boldsymbol{q}}_0 + \lambda_m(\overline{\boldsymbol{q}}^* - \overline{\boldsymbol{q}}_0)$ now will be a solution. This process of increasing λ is repeated until $\lambda = 1$ is reached, in which case the solution for the LCP has been obtained. It can be shown that λ can always be increased when the diagonal elements of M needed as a pivot are always positive. Moreover, if the matrix M belongs to class P, the Katzenelson algorithm will always find the unique solution (15,19).

As an example, consider a fairly simple network, consisting of a linear resistor in series with a nonlinear resistor that has a characteristic as defined in Fig. 2 and for which the model is given by Eq. (23). This network is excited by a voltage source E. The topological relation yields

$$E = Ri + v \tag{29}$$

For this network we intend to find the dc operating point for $E = 9$ V and $R = 4\ \Omega$. According to the theory given previously, we can write the complete network in terms of its input variable E and its output variable i by combining Eq. (23) with Eq. (29), yielding

$$\begin{aligned} &i + \begin{bmatrix} -\frac{1}{5} \end{bmatrix} E + \begin{bmatrix} \frac{3}{10} & -\frac{3}{10} \end{bmatrix} \boldsymbol{u} = 0 \\ &\boldsymbol{j} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} E + \begin{bmatrix} -4 \\ -4 \end{bmatrix} i + I\boldsymbol{u} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} \end{aligned} \tag{30}$$

where we leave out the complementary conditions for convenience. Because of the definition of the elements of the network, $(i, E_0) = (0, 0)$ is a solution of the network. However, we intend to obtain the dc operating point for $E_e = 9$ and therefore we may define the homotopy path as $E = E_o + \lambda(E_e - E_0) = \lambda 9$. We are now able to rewrite the state equation in Eq. (30) into a form similar to Eq. (27), yielding

$$\boldsymbol{j} = \begin{bmatrix} -\frac{1}{5} \\ -\frac{1}{5} \end{bmatrix} \lambda 9 + \boldsymbol{Iu} + \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Note that the LCP matrix is the identity matrix and thus of class P. Katzenelson's algorithm will always obtain a solution. Increasing λ to let $u_m > 0$ to prevent j_m from becoming negative results in $\lambda = 5/9$ for the first state equation. Let u_1 and j_1 interchange and Eq. (30) will be updated to

$$\begin{aligned} i + \left[-\frac{1}{2}\right] E + \left[-\frac{3}{2} \quad \frac{3}{2}\right] \boldsymbol{u} + \frac{3}{2} = 0 \\ \boldsymbol{j} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} E + \begin{bmatrix} -4 \\ 4 \end{bmatrix} i + \boldsymbol{Iu} + \begin{bmatrix} -1 \\ 2 \end{bmatrix} \end{aligned} \tag{31}$$

where u_1 and j_1 interchanged names also to achieve a model similar to Eq. (30). Note that $\lambda = 5/9$ means that $E = 5$, $i = 1$, and therefore $v = 1$, which is indeed a breakpoint of the nonlinear resistor characteristic (see Fig. 2). By further increasing E, the diode in the second branch of the subnetwork representing the nonlinear resistor starts to conduct as v increases. The complete network topology will now change and is described by the new mapping equation in Eq. (31). For this new situation Eq. (28) now yields

$$\boldsymbol{j} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \lambda 9 + \boldsymbol{Iu} + \begin{bmatrix} 5 \\ -4 \end{bmatrix}$$

from which it can be observed that we may not increase the homotopy parameter λ further. The alternative is to decrease this parameter and hope that we may increase it afterward to reach $\lambda = 1$. It can be proved that this extension to the original method of Katzenelson is allowed (19). Doing so, we obtain $\lambda = 4/9$ in the second state equation, which corresponds to the diode in third branch of the subnetwork representing the nonlinear resistor starting to conduct. Pivoting and updating the model results in

$$\begin{aligned} i + \left[-\frac{1}{5}\right] E + \left[\frac{3}{10} \quad -\frac{3}{10}\right] \boldsymbol{u} + \frac{3}{10} = 0 \\ \boldsymbol{j} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} E + \begin{bmatrix} -4 \\ -4 \end{bmatrix} i + \boldsymbol{Iu} + \begin{bmatrix} -1 \\ -2 \end{bmatrix} \end{aligned}$$

and the algorithm yields

$$\boldsymbol{j} = \begin{bmatrix} \frac{1}{5} \\ \frac{1}{5} \end{bmatrix} \lambda 9 + \boldsymbol{Iu} + \begin{bmatrix} \frac{1}{5} \\ -\frac{4}{5} \end{bmatrix}$$

from which it is clear that we may increase the homotopy parameter reaching $\lambda = 1$. We now have obtained the dc operating point of this network, $(i, E) = (3/2, 9)$, and the voltage over the nonlinear resistor is $v = 3$ V.

In the literature, an adaptation to the Katzenelson algorithm is presented in which a single homotopy parameter path is extended to a multiparameter path (20). The homotopy parameter may be complex. The advantage is that difficult points in the characteristic, such as the hysteresis curve, can be handled with more care than with the straightforward method. Another extension is treated in Ref. 17 that allows us to find the dc operating point of a network having a discontinuous behavior.

Over the years, several algorithms have been developed to solve the LCP and they can roughly be categorized into four groups:

1. *Homotopy Algorithms.* Besides the algorithm of Katzenelson, Lemke (21) and van der Panne (22) developed pivoting algorithms based on homotopy methods. The advantage of the latter two algorithms is that they are able to handle a larger class of LCP matrices than can Katzenelson, which is only guaranteed for class P problems. The price to be paid is a more complex algorithm, and therefore it is mainly the Katzenelson algorithm that is used in (PL) simulators.
2. *Iterative Algorithms.* These methods solve some equivalent multidimensional optimization problem. This optimization problem is most often quadratic (23). Equation (26) can be reformulated as minimizing $\frac{1}{2}\boldsymbol{x}^T M\boldsymbol{x} + \boldsymbol{q}^T\boldsymbol{x}$ under the condition that $\boldsymbol{x} \geq 0$, which yields a solution satisfying Eq. (26). The required solution can be obtained by applying efficient gradient search methods from the nonlinear optimization theory.
3. *Contraction Algorithms.* The algorithms in this class solve some equivalent nonlinear algebraic problem by iteration using, for example, contraction or Newton-Raphson iteration. One important member of this class is the modulus algorithm (8). This method will yield a polynomial solution algorithm for matrix M from a certain limited class such as positive definitive matrices.
4. *Polyhedral Algorithms.* These methods perform operations on the polyhedrons in which the domain space is divided by the collection of hyperplanes. We will discuss two algorithms of this class in more detail in the following section because this class of algorithms allows us to find all dc operating points of a network.

MULTIPLE DC OPERATING POINTS

In the previous section algorithms were discussed to obtain a single dc operating point of the electrical network. However, many circuits do have multiple operating points. We discussed how a solution algorithm (in this case Katzenelson) can be applied to solve a network of piecewise linear components (i.e., how to find a single operating point for a given excitation). In general, this means that using a homotopy method, we are able to find a single solution of a piecewise linear function starting from an initial condition. Determining all solutions would require trying all possible initial conditions, thus posing a severe drawback. The problem of finding all solutions of a system of piecewise linear (or, in general, nonlinear) equations is extremely complex. Because a piecewise linear function might have a solution in every region, any algorithm that claims to find all solutions must scan through all possible regions. The efficiency of an algorithm is therefore mainly determined by the efficiency with which it can remove regions that do not have a solution from the list of all regions. Finding all solutions of a piecewise linear function means solving

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} \tag{32}$$

To obtain all solutions of a piecewise linear function, we can use the brute force method. Knowing the linear map $\boldsymbol{y} = \overline{\boldsymbol{a}} + \overline{B}\boldsymbol{x}$ for each region, it is easily checked in which region the operating points are and what they are. However, this means solving 2^k linear equations, which can be a rather large number in general. Therefore, this is called the brute force method of solving Eq. (32). Hence it is worthwhile to develop methods that can reduce the computational effort of the task. For finding all solutions of a piecewise linear function, we must find an efficient way to exclude regions that do not contain a solution.

We will discuss several techniques that exploit some properties of the piecewise linear model to obtain rather efficiently all dc operating points of a network. To compare the methods, we will use one example throughout this section. We will consider the same network as in the previous section but with $R = 6\ \Omega$ and $E = 6$ V.

Exploiting the Lattice Structure

In 1982 Chua explored a special property of Eq. (6) to find all solutions in a more efficient way than the brute force method (24). This property is the fact that for functions described by Eq. (6) all regions in the domain space are separated only by horizontal and vertical hyperplanes. Notice that this property only holds for the one-level nested operator. Therefore, this method is not applicable for higher-order nesting of this operator, like in the model description based on higher-order base functions. Function f for our example is given by

$$f(v) = 0 = \frac{7}{6}v - \frac{7}{4} - \frac{3}{4}|v-1| + \frac{3}{4}|v-2| \tag{33}$$

which is obtained by combining Eq. (5) with the topological relation of Eq. (29). Now consider the domain space of f, which in this case is partitioned by 2 or, in general, by σ hyperplanes into 4 or 2^σ regions, respectively. Note their special property: They are parallel to one of the axis. Such a structure is called a lattice structure. For each region we have a linear map of f. We can also generate the partitioning of the image or the range space by applying map f on the regions. Because we are searching for the solutions of Eq. (32) or (33), we can see from the range space which regions must be considered, and they simply must contain the origin. Let x_1 be an arbitrary point in region R and let its image be y_1 in $\hat{R}$, the image of R. Consider also hyperplane H_k and its image $\hat{H}_k$:

$$\begin{aligned} H_k&: \langle \boldsymbol{\alpha}_k, \boldsymbol{x}\rangle - \beta_k = 0 \\ \hat{H}_k&: \langle \hat{\boldsymbol{\alpha}}_k, \boldsymbol{x}\rangle - \hat{\beta}_k = 0 \end{aligned} \tag{34}$$

If the origin is located in region $\hat{R}$, then

$$\operatorname{sgn}\big(\langle \hat{\boldsymbol{\alpha}}_k, 0\rangle - \hat{\beta}_k\big) = \operatorname{sgn}(-\hat{\beta}_k) \tag{35}$$

and this must be equal to

$$\operatorname{sgn}\big(\langle \hat{\boldsymbol{\alpha}}_k, \boldsymbol{y}_1\rangle - \hat{\beta}_k\big) \tag{36}$$

because $\boldsymbol{y}_1$ and the origin must lie on the same side of $\hat{H}_k$. This procedure must be repeated for all sides of region $\hat{R}$. If this so-called *sign test* fails on any of the boundaries of $\hat{R}$, then this region contains no solution of Eq. (32). Due to the sign test, we do not have to solve all linear equations, but only those for which we know in advance that they contain a solution. Therefore, this method is more elegant than the brute force method. In Ref. 24 Chua described an efficient implementation of the sign test.

Applying this technique to Eq. (33) yields the dc operating points $(i, v) = (\frac{6}{7}, \frac{6}{7})$, $(i, v) = (\frac{3}{4}, \frac{3}{2})$ and $(i, v) = (\frac{9}{14}, \frac{15}{7})$, which can be verified by adding the load line, defined by Eq. (29) to the characteristic in Fig. 2.

Separable Piecewise Linear Functions

Yamamura (25) developed a method that is based on the assumption that one considers the function $\boldsymbol{f}$ to be separable:

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{n} f^i(x) \tag{37}$$

where $f^i: R^1 \to R^n$. It can be shown that many practical resistive circuits exploit this property and hence this assumption is not too strict (26,27). Further, it is known that a piecewise linear approximation of a separable mapping can be performed on a rectangular subdivision. This means that if $\boldsymbol{f}$ was nonlinear, it is transformed into a piecewise linear function by approximating the function linearly within each rectangle. Hence a piecewise function will be the result. It also means that the following procedure results in an approximation of the exact solution: The finer the rectangular subdivision, the better the approximation solution of $\boldsymbol{f}$. If, however, $\boldsymbol{f}$ was already piecewise linear and, in particular, in accordance with Eq. (6), we can choose the subdivision such that it fits with the polytopes of the mapping. In case of Eq. (6) we choose the lattice structure as rectangular subdivision and the exact solutions will be obtained. If this is not possible, we can again approximate this piecewise linear function on a chosen rectangular subdivision following the procedure as if the function was nonlinear. So in this subsection we assume that $\boldsymbol{f}$ in Eq. (37) is either nonlinear or piecewise linear.

Let us subdivide the solution space into rectangular regions. To this purpose we define two vectors

$$\boldsymbol{l} = (l_1, l_2, \ldots, l_n)^T \text{ and } \boldsymbol{u} = (u_1, u_2, \ldots, u_n)^T \tag{38}$$

so that a particular n-dimensional rectangle is given by

$$R_i = \{\boldsymbol{x} \in R^n \,|\, l_i \le x_i \le u_i\}, \qquad i = 1, 2, \ldots, n \tag{39}$$

Then for this region R_i we define the following sign test:

$$\begin{aligned} &\sum_{i=1}^{n} \big[\max\{\hat{f}_j^i(l_i), \hat{f}_j^i(u_i)\}\big] \ge 0 \\ &\sum_{i=1}^{n} \big[\min\{\hat{f}_j^i(l_i), \hat{f}_j^i(u_i)\}\big] \le 0 \end{aligned} \qquad \mathrm{j} = 1, 2, \ldots, n \tag{40}$$

where $\hat{f}$ represents the linear approximation of $\boldsymbol{f}$ in the rectangle under consideration. Equation (40) means that in each rectangle only two function evaluations per region have to be performed. This is because the function within the rectangle is linear and hence the function evaluation on the boundaries of the rectangle provides enough information. For instance, if we consider the one-dimensional case, then Eq. (40) reduces to

$$\begin{aligned} &\max\{\hat{f}^1(l), \hat{f}^1(u)\} \geq 0 \\ &\min\{\hat{f}^1(l), \hat{f}^1(u)\} \leq 0 \end{aligned} \tag{41}$$

which means that at one boundary of the rectangle the function value is positive while at the other boundary the function value is negative. Indeed, somewhere within the boundary the function must pass the origin and hence a solution is obtained. If Eq. (40) does not hold for some j, the function does not possess a solution in that rectangle.

This test is very simple, simpler than the one proposed by Chua (24), where first the image of all boundaries must be computed. In the case of Yamamura, per region it requires only $2n(n-1)$ additions and $n(n+2)$ comparisons. After the sign test, we solve linear equations on the regions that passed the test. The problem with this method is that the test has to be applied on each rectangle. We can significantly reduce the number of tests by exploiting another property—namely, the sparsity of the nonlinearity. In general, each equation is nonlinear or piecewise linear in only a few variables and is linear in all other variables. Suppose that the function $\boldsymbol{f}$ is nonlinear in x_1 and linear in x_2; then we do not have to define a subdivision in R^2 but only in R. Now we can apply the same sign test of Eq. (40) to this structure, which has a complexity of a lesser degree than we had previously. We can show that the total complexity is on the order $O(n^3)$.

We can apply this technique to our example assuming that $\boldsymbol{f}$ is given by Eq. (33). Let us define the rectangular division as [0, 1], [1, 2], and [2, 3], which coincides with the lattice structure of Eq. (33). For the first rectangle Eq. (41) results in

$$\begin{aligned} &\max\{\hat{f}^1(0), \hat{f}^1(1)\} = \max\{-1, \tfrac{1}{6}\} \geq 0 \\ &\min\{\hat{f}^1(0), \hat{f}^1(1)\} = \min\{-1, \tfrac{1}{6}\} \leq 0 \end{aligned}$$

and therefore contains a solution of the network. In a similar way, we can observe that the other two rectangles fulfill the conditions, and working this out results in the three dc operating point as obtained previously.

Finding all Solutions Using Polyhedral Methods

In essence, these methods transfer the original problem into a form of the LCP and solve the new problem with very powerful methods. Any one-dimensional PL mapping can be written according to

$$\begin{aligned} x = {} & x_0 + x_{-\infty}\lambda^- + (x_1 - x_0)\lambda^+ \\ & + \sum_{k=2}^{n}(x_k - 2x_{k-1} + x_{k-2})\lambda_{k-1}^+ \\ & + (x_{+\infty} - 2x_n + x_{n-1})\lambda_n^+ \end{aligned} \tag{42}$$

$$\begin{aligned} &\lambda_j^+ - \lambda_j^- = \lambda^+ - \lambda^- - j, \quad j = 1, 2\ldots, n \\ &\lambda_j^+, \lambda_j^-, \lambda^+, \lambda^- \geq 0 \\ &\lambda^+ \cdot \lambda^- = 0, \lambda_j^+ \cdot \lambda_j^- = 0 \end{aligned} \tag{43}$$

where x_j, $j = 1, 2, \ldots, n$ represents a breakpoint in the characteristic and $x_{-\infty}$, $x_{+\infty}$ represents some points at the left-most and right-most segment (28). Equation (42) describes a PL mapping with the parameters consistent with the complementary conditions as given in Eq. (43). We did not mention this model description in the previous sections because it has no direct relation to an electrical network. The nonlinear resistor as defined by the network in Fig. 2 can be given in terms of Eqs. (42) and (43)

$$\begin{aligned} &\begin{bmatrix} i \\ v \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix}\lambda^- + \begin{bmatrix} -\frac{1}{2} \\ 1 \end{bmatrix}\lambda^+ + \begin{bmatrix} \frac{3}{2} \\ 0 \end{bmatrix}\lambda_1^+ \\ &\lambda_1^+ - \lambda_1^- = \lambda^+ - \lambda^- - 1 \\ &\lambda^+, \lambda^-, \lambda_1^+, \lambda_1^- \geq 0 \\ &\lambda_1^+ \cdot \lambda_1^- = \lambda^+ \cdot \lambda^- = 0 \end{aligned} \tag{44}$$

and the topological equation of Eq. (29) can be rewritten as

$$(-6 \quad -1 \quad 6)\begin{bmatrix} i \\ v \\ \alpha \end{bmatrix} \tag{45}$$

where α is a slack parameter. We can now substitute Eq. (44) into Eq. (45), yielding a system of the following form:

$$\begin{bmatrix} -9 & 0 & 2 & 7 & -1 \\ -1 & 1 & 1 & -1 & -1 \end{bmatrix}\begin{bmatrix} \lambda_1^+ \\ \lambda_1^- \\ \lambda^+ \\ \lambda^- \\ \alpha \end{bmatrix} \tag{46}$$

or, in general,

$$(M \quad N \quad -q)\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{z} \\ \alpha \end{bmatrix} = 0 \tag{47}$$

which in the literature is known as the *generalized linear complementary problem* (28,29) with $\boldsymbol{w}$, $\boldsymbol{z}$, $\alpha \geq 0$ and the complementary condition still valid. This set of equations can be solved using the modified Tschernikow method. The term *generalized* is used because the matrix is not of dimension $R^{n\times n}$, as in the LCP discussed previously, but can have any dimension, i.e. $R^{n\times m}$, $n \leq m$. Hence it can represent an underconstrained set of equations that indeed can possess more than one solution.

Tschernikow developed a method to find all solutions of the problem

$$A\boldsymbol{x} \leq \boldsymbol{b}, \qquad \boldsymbol{x} \in R^n, \qquad A \in R^{m\times n}, \qquad n \geq m \tag{48}$$

which in any case with the introduction of some slack variables can always be transformed into

$$B\boldsymbol{u} \leq 0, \boldsymbol{u} \geq 0 \qquad B \in R^{k\times p} \tag{49}$$

The solution space of Eq. (49) describes all nonnegative solutions of Eq. (48). The method starts to define a start tableau

$$T^1 = (T_1^1 | T_2^1) = \left[\begin{array}{ccc|ccc} 1 & & 0 & b_{11} & \cdots & b_{k1} \\ & \ddots & & \vdots & & \vdots \\ 0 & & 1 & b_{1p} & \cdots & b_{kp} \end{array}\right] \tag{50}$$

where T_1^1 is a unity matrix, forming a base in the p-dimensional space, and T_2^1 is composed by placing a row of Eq. (49) as column in Eq. (50). For each row in T_1^1 we define $S(i)$, $i = 1, 2, \ldots, p$ as the collection of columns in T_1^1 with a zero in row i. In a similar way, we define $S(i_1, i_2)$ as the collection with both zeros in i_1 and i_2. We now randomly choose a column j in T_2^1 with at least one nonzero element. We consider two rows, i_1, i_2, from the tableau with opposite sign in column j and consider the corresponding $S(i_1, i_2)$. If $S(i_1, i_2) \not\subset S(i)$, $i \neq i_1$, $i \neq i_2$, then the linear combination of rows i_1, i_2 such that a zero in column j is created is of importance. It is precisely this combination that generates a boundary in the solution space. Only on one side of this hyperplane, solutions of the problem do exist that are consistent with the space as defined in T_1^1 and the equation as defined by column j corresponding to row j of Eq. (49). Obviously, this new row must be introduced in the new tableau matrix. It must be clear that all rows having a zero or negative entry in column j are also transferred to the new tableau matrix. They automatically fulfill the inequality condition in Eq. (49) for axis j. In the same way tableau T^i can be found from T^{i-1}, and the procedure stops when all columns in the right part are treated or we end up with only columns in the right part, which are strict positive. In the latter case there does not exist a solution to the problem except the trivial solution. In the first situation we end up with the following tableau:

$$T^{\text{end}} = (T_1^{\text{end}} | T_2^{\text{end}}) = \left[\begin{array}{ccc|c} c_{11} & \cdots & c_{1p} & \\ \vdots & & \vdots & 0 \\ c_{t1} & \cdots & c_{tp} & \end{array}\right] \tag{51}$$

with the nonnegative solution for Eq. (49)

$$\boldsymbol{u} = \sum_{i=1}^{t} p_i c_i, \qquad \text{with } c_i = (c_{i1}, \ldots, c_{ip}) \tag{52}$$

and p_i a nonnegative parameter. The set $(c_1, \ldots, c_t)^T$ describes the corners of the convex solutions space. If the problem is written as

$$A\boldsymbol{x} = \boldsymbol{b}, \qquad \boldsymbol{x} \in R^n, \qquad A \in R^{m \times n}, \qquad n \geq m \tag{53}$$

as in Eq. (47), then only a small modification in the previously outlined procedure is needed. Only rows having a zero entry in column j are directly transferred to the new tableau matrix. For a detailed outline, we refer to the works of Tschernikow (30–32). For the generalized LCP, the procedure outlined previously has to be only slightly adapted: Now the complementary conditions must also be fulfilled, so after each generation of a new tableau we have to check these conditions. We simply check each row in the columns in the left part of the tableau matrix for whether the conditions are fulfilled or not. If not, the corresponding row must be removed from the tableau and we can generate a new tableau (29).

The start tableau in our example of Eq. (46) looks like

$$\left[\begin{array}{ccccc|cc} 1 & 0 & 0 & 0 & 0 & -9 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 & 0 & 7 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 \end{array}\right]$$

Taking the first column of the right-hand part, we can make two combinations of rows having opposite sign. All can be transferred to the next tableau, yielding

$$\left[\begin{array}{ccccc|c} 0 & 1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 9 & 0 & 0 & 7 \\ 7 & 0 & 0 & 9 & 0 & -16 \\ 0 & 0 & 1 & 0 & 2 & -1 \\ 0 & 0 & 0 & 1 & 7 & -8 \end{array}\right]$$

which finally yields (because many combinations do not fulfill the complementary conditions)

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 2 & 0 & 16 & 0 & 14 \\ 0 & 8 & 0 & 1 & 7 \end{bmatrix} \tag{54}$$

We now consider the first equation in Eq. (54), which tells us that $\lambda_1^+ = \frac{1}{2}(\alpha \equiv 1)$. Combining this with Eq. (44) leads to $(i, v) = (\frac{3}{4}, \frac{3}{2})$, which is indeed one of the dc operating points. In a similar approach, the other two operating points can be obtained from Eq. (54).

The outlined approach can be slightly modified to handle model descriptions as defined by van Bokhoven directly, leading to a broad class of problems that can be solved (33). Here first the transfer characteristic of each element is determined after which the topological relations are used to solve a set of equations similar to Eq. (47) but now being a pure LCP. The advantage of this method over the treated method is that restriction on the variables can be taken into account. This can be of interest when only solutions in a special subspace are of interest or when the network is extended with other components later. In that case, not the whole procedure must be restarted but only parts of it for the new added components. Obviously, this will save computational effort.

Polyhedral Methods and Linear Programming

For a long time the relation between LCP and linear programming (LP) has been known. Each LP problem can be transformed into an LCP using the duality property of the LP (29). On the other hand, it is possible to treat a piecewise linear network as a polyhedral function, which can then be solved using LP (34). We mentioned that the state equation describes a set of polyhedral regions in the space, called polytopes. For each polytope a linear relation describes the local behavior of the function. We can also combine these two relations when we treat the piecewise linear function as a polyhedral element. The polyhedral elements, in general, do not have a correspondence with a physical device, but they constitute a mathematical tool. Each polyhedral element consists of a set of polyhedral regions. For our nonlinear resistor in Fig. 2, one of the polyhedral sections would yield

$$\begin{bmatrix} i \\ v \end{bmatrix} = p_1 \begin{bmatrix} 0 \\ 0 \end{bmatrix} + p_2 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + p_3 \begin{bmatrix} \frac{1}{2} \\ 2 \end{bmatrix} = \sum_{i=1}^{3} p_i \boldsymbol{w}_i$$
$$p_1 + p_2 + p_3 = 1$$
$$p_1, p_2, p_3 \geq 0$$

describing the triangular area defined by the first two segments and a virtual line segment. We can use this description together with the topological equations to obtain a set similar to

$$\sum_{m=1}^{M} \sum_{i=1}^{K_m} p_i^m (w_i^m t_{km}) = r_k \text{ for } k = 1, \ldots, M$$
$$\sum_{i=1}^{K_m} p_i^m = 1, p_i^m \geq 0$$

where t and r define the topological relations, M is the number of polyhedral elements in the network, and K_m is the number of polyhedral regions per element. This system of linear equations and inequalities may be regarded as the constraints of a linear programming problem. The solution for each polyhedral region is a dc operating point for the original problem. To find all operating points implicates that, in principle, all polyhedral regions have to be solved. However, when we set up a "genealogical tree," a reasonable reduction in computation can be obtained. A certain node in this tree represents a specific polyhedral element. If a certain node does not contain a solution, some other nodes in the tree may be discarded. In Ref. 35 a detailed discussion can be found.

Comparison of the Methods

From the preceding discussions it follows that the method used first depends on the model descriptions used. When the network is described using an explicit model, then we can apply the method of Chua or Yamamaru, where the latter has the advantage of efficiency but demands separability of the network, which is not always the case. We know that explicit model descriptions are less powerful than implicit model descriptions and hence the polyhedral methods of Vandeberghe and Leenaerts can be applied to a larger class of problems. However, in situations where the network components are one-port elements, these methods are overkill with respect to the computation of the solutions. On the other hand, in the general case they are more powerful and do not require restrictions on the problem. The method of Pastore is not well accepted yet, mainly because it demands that all piecewise linear elements first have to be rewritten into polyhedral elements, which is not always a trivial task. Second, the problem must be solved using LP techniques, which does not well fit within a simulation environment. This latter drawback may also apply to the method of Yamamura. The methods of Chua, Vandeberghe, and Leenaerts fits very well within an existing simulation environment because they can handle the existing models directly and use techniques that are already available within the simulator to analyze the network in the time domain.

BIBLIOGRAPHY

1. L. O. Chua and S. M. Kang, Section-wise piecewise linear functions: Canonical representation, properties and applications, *Proc. IEEE,* **65**: 915–929, 1977.
2. S. M. Kang and L. O. Chua, A global representation of multidimensional piecewise linear functions with linear partitions, *IEEE Trans. Circuits Syst.,* **25**: 938–940, 1978.
3. L. O. Chua and A. C. Deng, Canonical piecewise linear representation, *IEEE Trans. Circuits Syst.,* **35**: 101–111, 1988.
4. G. Güzelis and I. Göknar, A canonical representation for piecewise affine maps and its application to circuit analysis, *IEEE Trans. Circuits Syst.,* **38**: 1342–1354, 1991.
5. C. Kahlert and L. O. Chua, A generalized canonical piecewise linear representation, *IEEE Trans. Circuits Syst.,* **37**: 373–382, 1990.
6. T. A. M. Kevenaar, D. M. W. Leenaerts, and W. M. G. van Bokhoven, Extensions to Chua's explicit piecewise linear function descriptions, *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.,* **41**: 308–314, 1994.
7. D. M. W. Leenaerts, Further extensions to Chua's explicit piecewise linear function descriptions, *Int. J. Circuit Theory Appl.,* **24**: 621–633, 1996.
8. W. M. G. van Bokhoven, *Piecewise Linear Modelling and Analysis,* Deventer, The Netherlands: Kluwer Technische Boeken, 1981.
9. W. M. G. van Bokhoven, Piecewise linear analysis and simulation, in A. E. Ruehli (ed.), *Circuit Analysis, Simulation and Design,* Amsterdam: North-Holland, 1986, pp. 129–166.
10. T. A. M. Kevenaar and D. M. W. Leenaerts, A comparison of piecewise-linear model descriptions, *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.,* **39**: 996–1004, 1992.
11. W. Kruiskamp and D. Leenaerts, Behavioral and macro modeling using piecewise linear techniques, *Int. J. Analog Integ. Circuits Signal Process.,* **10**: 67–76, 1996.
12. P. Veselinovic and D. M. W. Leenaerts, A method for automatic generation of piecewise linear models, *Proc. ISCAS,* Atlanta, GA, 1996, Part III, pp. 24–27.
13. L. O. Chua and A. Deng, Canonical piecewise linear modeling, *IEEE Trans. Circuits Syst.,* **33**: 511–525, 1986.
14. C. E. Lemke, On complementary pivot theory, in J. B. Rosen, O. L. Mangasarian, and K. Ritter (eds.), *Nonlinear Programming (Proceedings of a Symposium),* New York: Academic, 1968, pp. 349–384.
15. S. Kamardian, The complementarity problem, *Math. Program.,* **2**: 107–129, 1972.
16. J. Katzenelson, An algorithm for solving nonlinear resistor networks, *Bell Syst. Tech. J.,* **44**: 1605–1620, 1965.
17. P. Pejovic and D. Maksimovic, An algorithm for solving piecewise linear networks that include elements with discontinuous characteristics, *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.,* **43**: 453–460, 1996.
18. K. Kawakita and T. Ohtsuki, NECTAR 2, a circuit analysis program based on piecewise linear approach, *Proc. ISCAS,* Boston, 1975, pp. 92–95.
19. M. J. Chien and E. S. Kuh, Solving piecewise linear equations for resistive networks, *Int. J. Circuit Theory Appl.,* **4**: 3–24, 1976.
20. D. M. Wolf and S. R. Sanders, Multiparameter homotopy methods for finding DC operating points of nonlinear circuits, *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.,* **43**: 824–838, 1996.
21. C. E. Lemke, On the complementary pivot-theory, in G. B. Dantzig and A. F. Veinott, Jr. (eds.), *Mathematics of Decision Sciences,* Part I, Providence, RI: AMS, 1968, pp. 95–114.

22. C. van der Panne, A complementary variant and a solution algorithm for piecewise linear resistor networks, *SIAM J. Math. Anal.,* **8**: 69–99, 1977.
23. C. W. Cryer, The solution of a quadratic programming problem using systematic overrelaxation, *SIAM J. Control.,* **9**: 385, 1971.
24. L. O. Chua and R. L. P. Ying, Finding all solutions of piecewise-linear circuits, *Int. J. Circuit Theory Appl.,* **10**: 201–229, 1982.
25. K. Yamamura, Finding all solutions of piecewise-linear resistive circuits using simple sign tests, *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.,* **40**: 546–551, 1993.
26. K. Yamamura and M. Ochiai, An efficient algorithm for finding all solutions of piecewise-linear resistive circuits, *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.,* **39**: 213–221, 1992.
27. K. Yamamura, Exploiting separability in numerical analysis of nonlinear systems, *IEICE Trans. Fundam. Electron., Commun., Comput. Sci.,* **E75-A**: 285–293, 1992.
28. L. Vandeberghe, B. de Moor, and J. Vandewalle, The generalized linear complementarity problem applied to the complete analysis of resistive piecewise linear circuits, *IEEE Trans. Circuits Syst.,* **36**: 1382–1391, 1989.
29. B. de Moor, Mathematical concepts and techniques for modelling of static and dynamic systems, Ph.D. thesis, Katholieke Universiteit Leuven, Belgium, 1988.
30. S. N. Tschernikow, *Lineare Ungleichungen,* Berlin: VEB Deutscher Verlag der Wissenschaft, 1971 (translation from *Lineinye neravenstva,* 1968, by H. Weinert and H. Hollatz into German).
31. D. M. W. Leenaerts, Applications of interval analysis to circuit design, *IEEE Trans. Circuits Syst.,* **37**: 803–807, 1990.
32. D. M. W. Leenaerts, TOPICS, a contribution to design automation, Ph.D. dissertation, Department of Electrical Engineering, Technical University Eindhoven, The Netherlands, 1992.
33. D. M. W. Leenaerts and J. A. Hegt, Finding all solutions of piecewise linear functions and the application to circuit design, *Int. J. Circuit Theory Appl.,* **19**: 107–123, 1991.
34. V. Chvátal, *Linear Programming,* New York: Freeman, 1983.
35. S. Pastore and A. Premoli, Polyhedral elements: A new algorithm for capturing all equilibrium points of piecewise linear circuits, *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.,* **40**: 124–132, 1993.

DOMINE M. W. LEENAERTS
Technical University Eindhoven