INTERVAL ANALYSIS FOR CIRCUITS

Interval analysis is a novel tool for investigating linear and nonlinear lumped parameter circuits and systems. It is ideally suited to handle situations where the problem statement involves uncertainties in the form of intervals. Interval analysis offers a variety of computer methods for solving such problems. The new methods have better performance than the traditional (noninterval) methods.

Presently, there are two conceptually different approaches to investigating circuits and systems, the traditional approach and the interval approach. While the former is based on models and methods using classical mathematics, the latter appeals to the concepts and computational techniques of a branch of contemporary mathematics called interval analysis.

INTERVAL ANALYSIS

Interval analysis is a new and intensively developing area of applied mathematics. The first paper (1) in the field appeared in 1958 and the first monograph by Moore (2) was published in 1966. Originating as a tool to control propagation of round-off errors in computations on computers, interval analysis presently covers a variety of problems in computational mathematics which are difficult to solve by traditional approaches.

The basic concept in classical mathematics is the concept of a real number (a complex number can always, at least conceptually, be viewed as a pair of real numbers in a plane). In contrast, the basic concept in interval analysis is that of an interval. An interval is, geometrically, a bounded segment of the real line. Interval analysis studies mathematical relationships between intervals. Intervals are also called interval numbers. Interval numbers are generalizations of real numbers. Conversely, a real number can be viewed as a "degenerate interval" consisting of the real number itself. Similarly to real numbers, intervals can be arguments of functions called interval functions. The value of an interval function is an interval. One of the main objectives of interval analysis is to study the properties of interval functions and to seek efficient methods to evaluate them.

Various interval analysis methods have been developed for solving numerous problems in linear and nonlinear mathematical analysis. In fact, nowadays an interval counterpart exists for every significant problem and method encountered in classical mathematics [interested readers may refer to references (2–5)]. These methods have a number of appealing features (such as guaranteed accuracy, global convergence, etc.) which makes them attractive for various applications in science and engineering.

THE INTERVAL APPROACH

The overwhelming majority of the mathematical models now in use in circuit theory are based on the traditional approach. This approach is quite natural and satisfactory if the initial data about the electric circuit studied (parameters of passive elements, of voltage or current sources, etc.) are known exactly. In this case, each item of the input data is represented with reasonable accuracy as a real number. Because each real number is viewed geometrically as a point on a real line, all of the data related to the problem at hand is visualized as a point in a space of appropriate dimensionality. Therefore, for brevity of expression, a mathematical model based on such an approach is termed [as in (6)] a "point" model. Although intrinsically inaccurate, the point model is practically the best model for tackling problems in which the uncertainty of input data can be ignored. On the other hand, there are problems where the uncertainty in the data is significant and cannot be neglected. A typical example is the tolerance analysis problem where one is interested in the range of the variations of an output circuit characteristic (for instance, the variation of an output voltage) resulting from the tolerances of the circuit parameters. In circuit theory, the basic approach to handling such problems is to appeal to a probabilistic description of the problem and to apply a certain statistical method to solve it. This approach, however, is associated with the necessity to determine experimentally some probability law describing the probabilistic distribution of the input data. Another possibility is to resort to the theory of fuzzy sets. Once again, some statistical information is needed to describe the "fuzzyness" of the sets involved.

An alternative for treating electric circuits with inaccurate data is to apply the interval approach, that is, to employ the concepts and methods of interval analysis. Because interval analysis deals with intervals rather than points, it is ideally suited for handling circuit problems where initial data are allowed to take on values within some prescribed intervals. A mathematical model based on the interval representation of the input data is called, for brevity and in contrast to the point model, an "interval" model. Furthermore, a method for solving a particular applied problem which is based on an associated interval model and appeals to appropriate interval analysis techniques is called an interval method. Interval methods were introduced for the first time in the field of electrical circuits in the late seventies for tackling the tolerance analysis problem [see (7,8)]. They have since been applied to handling a number of problems arising in the domain of electric circuit analysis [see (6–8,11,12,15–17)].

AREAS OF APPLICATION

At the present stage of their development, interval methods have covered the following two major areas of application: (1) robust analysis of linear circuits (and systems); (2) analysis of nonlinear circuits with exact data. The former topic is characterized by uncertain parameters which take on values within certain domains and most often these domains are given as intervals. The objective of the analysis is to check whether the circuit investigated is robust against the parameter variations, that is, to assess whether a certain output characteristic of the circuit remains within prescribed bounds for all possible variations of the uncertain parameters. More specifically, interval methods have proved successful in solving the following robustness problems.

Tolerance Analysis

In this problem the output characteristic is typically the dc or rms value of a voltage (current), and it is necessary to determine the voltage range under all admissible variations of the parameters, that is to determine the tolerance on the output characteristic given the tolerances on the input parameters. Two statements of the tolerance problem are encountered: (1) worst-case (deterministic) and (2) probabilistic statement.

In the former case, each parameter varies independently from the rest within a given interval. Thus the tolerance on the output variable accounts for the worst possible combinations of the admissible values of the input parameters.

In the latter case, the highly improbable combinations are eliminated by introducing a suitable probabilistic law of distribution which takes into account the interdependence among the parameter values.

Both tolerance problems are formulated as an associated global optimization problem. The latter problem is solved by various interval methods: zero-order method (using no derivatives of the functions involved), first- and second-order methods (using first- and second-order derivatives, respectively). The worst-case tolerance problem is also formulated as a specific system of linear equations with independent or dependent interval coefficients. This mathematical model proves more efficient than the global optimization formulation in the case of electric circuits of increased size. Exact solution of the dc tolerance problem and approximate solutions to the ac tolerance problem are thus derived.

Robust Stability

Now the objective of the analysis is to establish that the circuit investigated remains stable for all admissible independent parametric variations given as intervals. This basic problem is extended to encompass various alternative formulations in which certain stability margins are introduced. Two approaches to treating the robust stability problem are known. According to the first, the stability of the circuit investigated is assessed by an associated characteristic polynomial whose coefficients, in the general case, are nonlinear functions of the interval parameters. The second approach is associated with assessing the stability of a corresponding interval matrix (a matrix whose elements are intervals). Extending some known results on stability for exact data circuits, necessary and sufficient conditions and simpler sufficient conditions are thus obtained for checking the stability, instability, or stability margin of linear circuits and systems with interval parameters.

Transient Analysis

This application area is concerned with transient analysis of linear circuits with uncertain (interval) parameters. In fact, the robust problem considered is a dynamic generalization of the static, worst-case tolerance analysis problem. Unlike the latter problem, the input interval data may, however, include not only the circuit parameters, but also input exitations and initial conditions. Each combination of these input parameters determines a corresponding output variable which is a function of time. In the most general case, the objective of the analysis is to verify whether the set of all output variable functions related to the set of admissible input parameters remains within a given preset "funnel." Various special cases are also possible. A well-known example is the problem where the output variable should not exceed some prescribed threshold value (typically, the tolerated overshoot of the dynamic system investigated) under all admissible parametric variations (therefore, in control engineering literature, the transient analysis problem is usually called the robust performance problem). A similar problem arises in setting relay protections where the relay should not react to all responses of the circuit protected caused by normal parametric variations but should do so under abnormal conditions. Once again, determining the maximum value of the corresponding circuit response under all possible parametric changes is of paramount importance.

A basic assumption in solving the robust transient (performance) problem is the assumpton that the linear circuit investigated is robustly stable. This can be checked by an appropriate method for analyzing robust stability.

Various methods for exact or approximate solution of the transient analysis problem have been proposed. In the simplest case, the relationship between the input parameters and the output variable must be available in a closed explicit form which is possible only for circuits of low complexity (circuits whose transients are described by a differential equation of first or second order). In this case, the transient analysis problem is solved exactly. For circuits of higher complexity, two alternative formulations have been suggested. The former formulation is in the frequency domain whereas the latter is in the time domain. Several methods for exact and approximate solutions have thus been developed.

Interval methods have also proved a reliable and efficient tool for analyzing and simulating nonlinear circuits. For the time being, they have mainly been applied to treating circuit analysis problems with exact data.

Nonlinear Circuit Analysis

Nonlinear Resistive Circuits. Global analysis (locating all operating points) of nonlinear circuits is one of the most challenging nonlinear problems. The interval approach has made possible the complete solution of the global analysis problem relative to the class of nonlinear resistive circuits. This problem has two versions depending on whether the nonlinear resistors involved are modeled by piecewise-linear (PWL) functions or by continuously differentiable (CD) functions. The traditional methods solve the former problem only in the case where the resistive circuit equations are written in the so-called hybrid-representation form (9). Traditional methods do not guarantee the location of all operating points for circuits whose nonlinear elements are modeled by CD functions. In contrast, existing interval methods find all operating points infallibly within prescribed accuracy in the general case, where the resistive circuit is described by a system of nonlinear equations of general form, and in the case of equations of the hybrid form.

Nonlinear Dynamic Circuits. This class of circuits presents a vast domain for interval analysis applications. Presently, the interval approach has been employed to solve the following two problems. First, a global analysis problem of finding all the periodic steady states of a given period arising in a nonlinear electric circuit has been addressed. A method for solving this problem in the case of circuits of low dimension (described by nonlinear differentiable equations of up to second or third order) has been suggested. Second, the challenging problem of establishing the uniqueness of a periodic steady state has also been considered. A new result has been obtained for a special case of circuits for which the system of nonlinear differential equations describing the circuit is of the so-called separable form (6). A sufficient condition for uniqueness of the periodic steady state in this class of circuits is suggested which reduces the original uniqueness problem to that of checking the stability of an associated interval matrix. The latter problem is efficiently solved by an approximate interval method.

VIRTUES AND DRAWBACKS OF THE INTERVAL APPROACH

Interval methods have a number of appealing features. One of their fundamental virtues is that, unlike the traditional methods where each computed output value is obtained as a real number, they provide each output result as an interval. The interval contains the result sought, thus guaranteeing infallible bounds on the true value of the respective output value. Using the so-called machine arithmetic, interval methods automatically account for roundoff errors when implemented by computer. For this reason they are often termed self-validating methods. Interval methods are more reliable than their noninterval counterparts. This is particularly true for the class of iterative methods used to solve nonlinear problems. Interval iterative methods always converge globally in a finite number of steps whereas their noninterval counterparts sometimes do not. Also, natural stopping criteria exist for interval iterations. One simply iterates until the bounds are sufficiently sharp (the resulting interval is narrow enough) or no further reduction of the interval bounds is possible. The latter occurs when rounding errors prevent further accuracy improvement. Interval methods solve nonlinear problems globally. Thus, these methods find all solutions of a set of nonlinear equations is a given rectangular region (a box). Similarly, they find the global optimum (s) of an (unconstrained or constrained) optimization problem in a finite number of steps with guaranteed accuracy. Using traditional (point) methods, one faces the risk of terminating the computational process prematurely before globality is reached or continuing it uselessly in the hope of finding new solutions or better local optima (long after globallity has actually been reached). Interval methods require shorter computational time in most of the cases studied so far. However, the transcendent virtue of the interval approach is that it solves problems which were previously insoluble. For instance, before the use of interval methods, it has been impossible to find with certainty all operating points in resistive nonlinear circuits described by CD functions.

On the other hand, programming and using interval methods is presently less convenient than traditional methods. Indeed, all the interval operations involved in the method used have to be programmed individually for every problem being solved by the developer or user of the method. This lack of convenience, however, is avoidable. High-level algorithmic languages already exist (e.g., Pascal SC, Fortran SC, Ada, C^{++}), which permit intervals to be declared as a special data type. Special routines to do the interval arithmetic, however, are also needed, as are codes to evaluate the elementary transcendental functions, etc. These facilities are presently available for only a few languages restricted to only a few computers. Good interval software for various applied problems is often available. Portable codes are, however, comparatively rare.

INTERVAL ARITHMETIC

Interval Numbers

Let a, b be real numbers and let $X = [a, b]$ denote a closed bounded interval on the line of real number x, that is, $a \leq x \leq b, a \leq b$, and a, $b \leq \infty$. In interval analysis, such intervals are called interval numbers, and the two terms "interval number" and "interval" are used interchangeably. Thus, an interval number X is a closed bounded compact set of real numbers. To distinguish interval numbers from real numbers, the former are designated most often by capital letters whereas lower case letters are retained for real numbers. Lower-case letters with superscript I are also employed to denote intervals explicitly whenever needed to avoid ambiguity. Furthermore, if X is an interval, its lower (left) endpoint is denoted by $\underline{x}$ or x^{L} and its upper (right) endpoint by $\overline{x}$ or x^{R}. An interval can be regarded in two different ways, either as a set of real numbers or an ordered pair of two real numbers x^{L} and x^{R}. However, from a computational point of view, the latter representation offers great advantages over the former because it permits reducing operations with interval numbers to operations involving only their endpoints, thus avoiding the more cumbersome operations with sets.

An interval X is called degenerate if $x^L = x^R$. The interval number is a generalization of the real number. Indeed, in terms of interval analysis, any real number x is considered a degenerate interval $x = [x, x]$. Two intervals $X = [a, b]$ and $Y = [c, d]$ are equal if and only if (iff) their corresponding endpoints are equal, that is, $X = Y$ iff $a = c$ and $b = d$. Intervals are ordered in the following way: $X < Y$ iff $b < c$. A useful relationship for intervals is the set inclusion: $X \subseteq Y$ iff $a \geq c$ and $b \leq d$.

The width of an interval X is defined as the real number $w(X) = b - a$. It is easily seen that $w(X) \leq w(Y)$ when $X \subseteq Y$. The midpoint (or center) of X is the real number $m(X) = (a + b)/2$. Let $r = w(X)/2$ and $m = m(X)$. An interval X is defined either by specifying its endpoints a and b or, equivalently, in the form

$$X = m + [-r, r] = [m - r, m + r] \quad (1)$$

In interval analysis, the quantity r is called the radius of the interval. In technical literature, r is termed "tolerance" and is usually given in percents of the "nominal value" $m(X)$.

Interval numbers are ordered as one-dimensional or two-dimensional arrays to form interval vectors $X = (X_1, \ldots, X_n)$ or interval matrices $A = \{A_{ij}\}$, $i, j = 1, \ldots, n$, respectively.

The relationships of equality (=), inclusion ($\subseteq$) and ordering ($\langle$ on $\rangle$) introduced for interval numbers also remain valid for interval vectors and interval matrices iff they are extended to all components. Thus, the notation $X \subseteq Y$, where X and Y are interval vectors, means that $X_i \subseteq Y_i$, $i = 1, \ldots, n$, X_i and Y_i being the components of X and Y, respectively. The midpoint (center) $m(X)$ of an interval vector X is defined by the real vector $m(X) = (m(X_1), \ldots, m(X_n))$. The width of X, however, is given by the real number $w(X) = \max\{w(X_i), i = 1, \ldots, n\}$.

Interval Arithmetic Operations

Let $+, -, \cdot, /$ denote the operations of addition, subtraction, multiplication and division, respectively, over real numbers. Furthermore, let $*$ denote any one of these operations for the real numbers x and y. Then the corresponding operation for the interval numbers X and Y is defined as the set

$$X * Y = \{x * y: \quad x \in X, \quad y \in Y\} \quad (2)$$

Thus, the set $X * Y$ resulting from the operation considered contains every possible number which can be formed as $x * y$ for each $x \in X$ and each $y \in Y$. A fundamental requirement for $X * Y$ is to be an interval, that is the set $X * Y$ must be a bounded set. This is always true for the first three operations. Then the definition given by Eq. (2) produces the following rules for generating the endpoints of $X * Y$ from the endpoints of the two intervals $X = [a, b]$ and $Y = [c, d]$:

$$\begin{aligned} X + Y &= [a + c, b + d] \\ X - Y &= [a - d, b - c] \\ X \cdot Y &= [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)] \end{aligned} \quad (3)$$

The endpoints of the product are computed in a less expensive way if the signs of the endpoints of X and Y are taken into account [see (2–6)]. For brevity, the dot in the notation of the product is often dropped. The operation of division is possible only if Y is an interval not containing zero. In this case

$$1/Y = [1/d, 1/c]\ (0 \notin Y), \quad X/Y = X \cdot (1/Y)\ (0 \notin Y) \quad (4)$$

The restriction $0 \notin Y$ is removed if the so-called extended interval arithmetic [suggested by Hansen (5)] is used where intervals are unbounded.

Properties of Interval Arithmetic

If X and Y are degenerate intervals, then Eqs. (3), (4) reduce to the ordinary arithmetic operations over real numbers. Thus, interval arithmetic is a generalization of real arithmetic. Therefore, it is normal to expect that the properties of interval arithmetic are similar to those of real arithmetic, which is really the case. However, there are several striking dissimilarities that are stressed here. It is important to underline that, unlike real arithmetic, $X - X \neq 0$ and $X/X \neq 1$ when $w(X) > 0$. Indeed, $X - X = w(X)[-1, 1]$ and $X/X = [a/b, b/a]$ for $X > 0$ or $X/X = [b/a, a/b]$ for $X < 0$. Another interesting property of interval arithmetic is the fact that the distributive law

$$X(Y + Z) = XY + XZ \quad (5)$$

does not always hold. For example, $[0, 1](1 - 1) = 0$ whereas $[0, 1] - [0, 1] = [-1, 1]$. We do, however, always have the following inclusion:

$$X(Y + Z) \subseteq XY + XZ \quad (6)$$

The property given by Eq. (6) is called subdistributivity. It is to be stressed that, as seen from Eq. (6) and the previous example, $w(X(Y + Z)) \leq w(XY + XZ)$. Therefore, it is always advantageous to use the factored form $X(Y + Z)$ rather than the expression $XY + XZ$ because the former form leads, in general, to a narrower resultant interval. It is proved that Eq. (5) remains true in several special cases [see (2–6)].

Another important property of interval arithmetic is inclusion monotonicity. It means that, if $X \subseteq Z$, $Y \subseteq W$, then $X + Y \subseteq Z + W$, $X - Y \subseteq Z - W$, $XY \subseteq ZW$, $X/Y \subseteq Z/W$ (if $0 \notin W$ in the division formula). Inclusion monotonicity follows directly from the definitions of the interval arithmetic operations.

The arithmetic operations defined by Eqs. (3) and (4) are called exact interval arithmetic operations. However, when implementing these operations on a computer, we commit errors because of round-off. Therefore, we have to take special measures so that the machine-computed interval result always contains the exact interval result. When computing with interval arithmetic, if a left endpoint is not machine representable it is rounded to the nearest arithmetically smaller machine number. A right endpoint is rounded to the nearest arithmetically larger machine number. This is termed outward rounding. In what follows, various interval methods are presented. For simplicity, only exact interval arithmetic is used although the actual computer implementation of these methods, naturally, require machine interval arithmetic.

INTERVAL FUNCTIONS

An interval function is an interval-valued function of one or more interval arguments. The interval function F of interval variables $X_1, \ldots, X_n$ is denoted $F(X_1, \ldots, X_n)$, and F transforms the set of intervals $X_1, \ldots, X_n$ into the interval function value Y, that is, $Y = F(X_1, \ldots, X_n)$. An interval function is said to be inclusion monotonic if $X_i \subseteq Y_i$, $i = 1, \ldots n$, implies $F(X_1, \ldots, X_n) \subseteq F(Y_1, \ldots Y_n)$. It follows from Eq. (2) that interval arithmetic is inclusion monotonic, that is, if $X_i \subseteq Y_i$, $i = 1, 2$, then, $(X_1 * X_2) \subseteq (Y_1 * Y_2)$. The inclusion monotonicity is a property often used in interval computations.

Interval Extensions

Interval functions are engendered by real functions. The corresponding interval function is called an interval extension of the real function. More specifically, if $F(X_1, \ldots X_n)$ is an interval extension of $f(x_1, \ldots x_n)$, then F reduces to f when all arguments X_i become real variables, that is, $F(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$. Consider, for example, a rational real function of real variables (a function whose value is defined by a finite sequence of real arithmetic operations over its arguments). We obtain an interval rational function F engendered by the real function f if we replace the real variables in f by corresponding intervals and the real arithmetic operations by their interval counterparts. The resulting interval function F is termed a natural interval extension of f. Similarly, we obtain natural interval extensions of any real functions (containing irrational terms).

It should be stressed that different expressions of one and the same real function give rise to different interval extensions. For example, let $f(x) = x(1 - x) = x - x \cdot x$. The natural extension for the first expression $f(x) = f_1(x) = x(1 - x)$ is $F_1(X) = X(1 - X)$ whereas, for the second expression $f(x) = f_2(x) = x - x \cdot x$, the corresponding natural extension is $F_2(X) = X - X \cdot X$. Now, if we compute $F_1(X)$ and $F_2(X)$ for $X = [0, 1]$, $F_1([0, 1]) = [0, 1]$ whereas $F_2([0, 1]) = [-1, 1]$. Obviously, $F_1(X) \neq F_2(X)$. Moreover $F_1(X) \subset F_2(X)$. This example shows that, for polynomials, the nested form $A_0 + X[A_1 + X(A_2 + \ldots XA_n) \ldots]$ is never worse and is usually better than the sum of powers $A_0 + A_1X + A_2XX + \ldots$ because of subdistributivity. Henceforth, whenever we refer to the natural interval extension of a real function, we shall assume that an expression of the function has already been chosen.

Mean-Value Form

The mean-value form is a particular form of interval extension applicable to any function $f(x_1, \ldots x_n)$ with continuous first derivatives. Let $X = (X_1, \ldots, X_n)$ denote an interval vector, and let $m = m(X)$ be its center. By the mean-value theorem, for any $y \subset X$,

$$f(y) = f(m) + \sum_{j=1}^{n} \frac{\partial f}{\partial x_j}(\xi)(y_j - m_j), \quad \xi \in X$$

If $F'_j(X)$ denotes the (natural) interval extension of $\partial f/\partial x_j(x)$ for $x(x_1, \ldots, x_n) \in X$, then the interval function

$$F_{\mathrm{MV}}(X) = f(m) + \sum_{j=1}^{n} F'_j(X)(X_j - m_j) \tag{7}$$

is called the mean-value form extension of f on X. The mean-value form is inclusion monotonic if the functions $F'_j(X)$, $j = 1, \ldots n$, are inclusion monotonic.

Range

The set of real points (i.e., vectors) x belonging to an interval vector X with components X_i, $i = 1, \ldots, n$, form an n-dimensional parallelepiped with sides parallel to the coordinate axes. This is why an interval vector is often referred to as a box. Another important concept closely related to the interval extension of a real function is the range of the function over a box. The range $f(X)$ of f over X is an interval defined by the set $f(X) = \{f(x): x \in X\}$. Obviously, the range is the union of all function values $f(x)$ for all x from X. Enclosing the range of a multivariate function by an interval is a fundamental problem encountered in numerous applications. It is a standard problem in the field of robustness analysis. It is proved that

$$f(X) \subseteq F(X) \tag{8}$$

where $F(X)$ is an inclusion monotonic interval extension of $f(x)$. Consider the following example. Let $f(x) = x(1 - x)$. The range of $f(x)$ over $X = [0, 1]$ is easily computed to be $f([0, 1]) = [0, 0.25]$. From the previous example, $F([0, 1]) = F_1([0, 1]) = [0, 1]$. Thus, $f([0, 1]) \subset F([0, 1])$.

The inclusion in Eq. (8) is one of the basic results of interval analysis. We find infallible bounds on the range of $f(x)$ over X by just computing the interval extension $F(X)$. However, the bounds thus found, typically, are not very sharp, especially when the box X is fairly large. One of the central problems in interval analysis is finding a good estimate of $f(X)$ with a reasonable amount of computation. In two special cases, the range is found in a straightforward way [see (2–5):

1. The function f is a monotonic (in the classical sense) function of one variable for $x \in \mathrm{X} = [a, b]$. For monotonically increasing functions, such as $\sqrt{x}$, exp x, log x etc., $f(X) = [f(a), f(b)]$. For a monotonically decreasing function, $f(X) = [f(b), f(a)]$.
2. The function f is a multivariate function such that each variable x_i occurs not more than once and to the first power. Then $f(X)$ is found directly by computing $F(X)$ only once (provided no division by an interval containing zero occurs) because $F(X) = f(X)$ in this case.

Excess

In general, the interval extension is a wider interval than the range. To measure the closeness of $F(X)$ to $f(X)$, we use the so-called excess $E[F(X)] = w[F(X)] - w[f(X)]$. Let $d = w(X)$. It has been proved that, if $F(X)$ is a natural interval extension of a function f, then

$$E[F(X)] = 0(d) \tag{9}$$

(the above symbol means that E becomes proportional to d as d tends to zero). If the mean-value form $F_{\mathrm{MV}}(X)$ is used as the interval extension of f, then

$$E[F_{\mathrm{MV}}(X)] = 0(d^2) \tag{10}$$

(in this case E is second order in d). It should be noted that Eqs. (9) and (10) are asymptotic. They are useful expressions only when d is small. For large width d of the box $X = (X_1, \ldots, X_n)$ and large number n of interval arguments X_i, the excess is significant. This is a drawback of the interval analysis approach which is referred to as overestimation. Nowadays, there are a number of methods to reduce the excess. However, they usually involve numerous evaluations of $F(X^{(v)})$ for different subregions $X^{(v)}$ of X and sometimes are prohibitively expensive.

INTERVAL METHODS FOR LINEAR EQUATIONS

Consider the system of linear equations

$$Ax = b \tag{11}$$

where A and b is a real $(n \times n)$ matrix and a real vector, respectively. In many applications (tolerance analysis is a typical example), the elements of A and/or the components of b are not precisely known. If we know an interval matrix A^{I} bounding A and an interval vector B bounding b, we can replace the system in Eq. (11) by the family of linear systems

$$Ax = b, \quad A \in A^{\mathrm{I}}, \quad b \in B \tag{12}$$

For brevity, Eq. (12) is written in the form

$$A^{\mathrm{I}}x = B \tag{13}$$

In what follows, we assume that A^{I} is a regular matrix. A^{I} is regular if each $A \in A^{\mathrm{I}}$ is nonsingular. The solution set of Eq. (13) is the set

$$S = \{x\colon \quad x = A^{-1}b, \quad A \in A^{\mathrm{I}}, \quad b \in B\}$$

This set has a very complicated shape and therefore is impractical to use. Instead, it is common practice to settle for an interval vector X which contains S. In some cases we would, however, like to find the narrowest interval vector $\tilde{X}$ that still contains S. The vector X is called the interval solution of Eq. (13) whereas $\tilde{X}$ is called the optimal solution. Figure 1 shows a set S and the corresponding optimal solution $\tilde{X}$ for the case where $n = 2$. It should be stressed that $\tilde{X}$ (and moreover X) is not a solution in the classical sense. Indeed, if we replace x by $\tilde{X}$ in Eq. (13) and perform the interval multiplications and additions, the resulting interval vector $Y = A^{\mathrm{I}}\tilde{X}$, in general, is not equal to B.

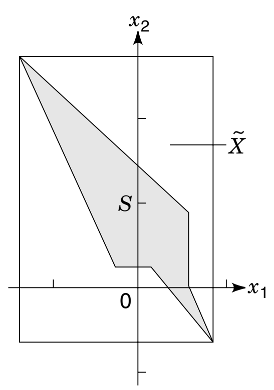

Figure 1. A two-dimensional example illustrating the smallest possible inclusion of the solution set S in the optimal interval solution $\tilde{X}$.

Interval Solution

A variety of methods exist for solving Eq. (13). Only three such methods are considered here.

Gaussian Elimination. There are several variants of a method for solving linear equations with exact data which are labelled Gaussian elimination. An interval version of any one of them is obtained from a standard one (using ordinary real arithmetic) by simply replacing each ordinary arithmetic step by the corresponding interval arithmetic step. If the coefficient matrix A and the right-hand side b are real (noninterval), then the interval version of Gaussian elimination simply bounds rounding errors. If the elements of the coefficient matrix and the right-hand side vector are intervals, then the solution vector X bounds the solution set S. Unfortunately, the bounds tend to widen rapidly because of accumulated overestimation at each step of the method. Thus, the solution obtained is generally far from sharp.

Preconditioning. To improve the performance of the Gaussian elimination, a technique suggested by Hansen (5), called preconditioning, is often used. The improvement is substantial for relatively small widths of A^{I} and B. Let A_c denote the center of A^{I}. First we compute (using, for example, real Gaussian elimination) an approximate inverse L of A_c. Then we multiply both sides of Eq. (13) by L to get the preconditioned set of equations

$$M^{\mathrm{I}}x = R \tag{14}$$

with $M^{\mathrm{I}}x = LA^{\mathrm{I}}$ and $R = LB$. Now Eq. (14) is solved by the interval Gaussian elimination method. The preconditioning method involves, however, about six times as many operations as ordinary interval Gaussian elimination.

The Gauss–Seidel Iteration. If a crude initial enclosure $X = (X_1, \ldots, X_n)$ for S is known, it is possible to solve the modified Eq. (14) more efficiently. The ith equation of Eq. (14) is

$$M_{i1}x_1 + \ldots + M_{in}x_n = R_i$$

Solving for x_i and replacing the other components by their interval bounds, we obtain the new bound

$$Y_i = \left(R_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} M_{ij}X_j\right) / M_{ii} \tag{15}$$

The intersection

$$X_i' = X_i \cap Y_i \tag{16}$$

now replaces X_i. We successively computer X_i' using Eqs. (15) and (16) with $i = 1, \ldots, n$. The intersection, given by Eq. (16), is done at each step so that the newest bound is used in Eq. (15) for each variable with $j < i$. It should be noted that extended interval arithmetic must be used to encompass the case where M_{ii} contains zero.

The Optimal Solution

Gaussian elimination and Gauss–Seidel iteration yield the optimal solution $\tilde{X}$ only in some rather special cases [see (2–6)]. A general method for finding $\tilde{X}$ has been suggested by Rohn (10). Analogous due to Eq. (1), A^I and B are written as

$$A^I = [A_c - \Delta, A_c + \Delta], \quad \Delta \geq 0$$
$$B = [b_c - \delta, \quad b_c + \delta], \quad \delta \geq 0$$

where A_c and b_c are the center of A^I and B, respectively, and Δ and δ are their radii (here and later on, the sign for equality, inequality, inclusion or absolute value relating vectors or matrices is meant componentwise). Let W denote the set of all n-dimensional vectors whose components are either $+1$ or -1. Thus, W consists of 2^n vectors. For each $w \in W$, let T_w denote a diagonal matrix whose diagonal is w. To each n-dimensional real vector y, we assign the vector sign y whose components are $+1$ if $y_i \geq 0$ and -1 otherwise. Hence $y \in W$. For any w, $z \in W$, we form $A_{wz} = A_c - T_w \Delta T_z$, $b_w = b_c + T_w \delta$. Consider the system

$$A_{wz} x = b_w \tag{17a}$$

$$T_z x \geq 0 \tag{17b}$$

It has a unique solution $x^w = (x_1^w, \ldots, x_n^w)$ for every w. The system given by Eq. (17) is to be solved 2^m times (once for each $w \in W$). Then it is proved that the endpoints of the optimal solution components $\tilde{X}_i$ are found as follows:

$$\begin{aligned} \tilde{X}_i^L &= \min\{x_i^w, \quad w \in W\} \\ \tilde{X}_i^R &= \max\{x_i^w, \quad w \in W\} \end{aligned} \tag{18}$$

The solution x^w is found using the following algorithm.

Sign-Accord Algorithm. The sign-accord algorithm comprises the following steps:

0. For a given w find $z = \text{sgn}(A_c^{-1} b_w)$
1. Solve the system of linear equations $A_{wz} x = b_w$.
2. If $T_z x \geq 0$, terminate. In this case $x_w := x$ (the symbol := has the usual meaning of assignment). Otherwise go to the next step.
3. Find the index k for which $z_j x_j < 0$ for the first time. Let $z_k = -z_k$, and return to Step 1.

It is proved by Rohn (10) that the sign-accord algorithm terminates in a finite number of iterations. Very often, if A^I is narrow enough, it actually converges in only one iteration.

INTERVAL METHODS FOR NONLINEAR EQUATIONS

Nonlinear Equations of One Variable

Let f be a continuously differentiable scalar function of a single variable x. We consider the problem of finding all the zeros of $f(x) = 0$ in a given interval X_0. Among the various interval methods suggested for solving this problem, the interval modification of the Newton method is currently superior to its rivals.

The Interval Newton Method. From the mean-value theorem

$$f(x) - f(y) = f'(\xi)(x - y) \tag{19}$$

where ξ is some point between x and y. If y is a zero of f, then $f(y) = 0$ and from Eq. (19)

$$y = x - f(x)/f'(\xi) \tag{20}$$

Let X be an interval containing both x and y. Then $\xi \in X$ and hence $f'(\xi) \in F'(X)$ where F' is some interval extension of f'. Denote $N(x, X) = x - f(x)/F'(X)$. It follows from Eq. (20) that, if y is a zero of f in X, then $y \in N(x, X)$ and hence it is also in the intersection $X \cap N(x, X)$. The interval Newton method is based on this fact and has the following algorithm for finding a zero of f in X:

$$N(x_k, X_k) = x_k - f(x_k)/F'(X_k) \tag{21a}$$

$$X_{k+1} = X_k \cap N(x_k, X_k), \quad k \geq 0 \tag{21b}$$

with $x_k \in X_k$. Usually, x_k is taken as the center of X_k. The above algorithm was derived by Moore (2) for the case where $0 \notin F'(X_0)$. It was extended by Hansen (5) to allow $0 \in F'(X_k)$. In the latter case, $N(x_k, X_k)$ is computed by extended interval arithmetic. Then X_{k+1}, as computed from Eq. (21), consists of two intervals. Whenever this occurs, one of these is stored in a list L and processed later. This algorithm is called the extended interval Newton algorithm [a detailed description of the steps of the algorithm is given in (5)].

Properties of the Extended Algorithm. We list some of the basic properties of the extended-interval Newton algorithm which illustrate its reliability and efficiency.

1. The algorithm is globally convergent. Every zero of f in the initial interval X_0 is always found and correctly bounded within a given accuracy ϵ after a finite number of iterations (if f and f' have a finite number of zeros in X_0).
2. If there is no zero of f in X_0, the algorithm computationally proves this fact after a finite number of iterations (when the intersection in Eq. (21b) becomes empty and the list L contains no further subintervals to be processed).
3. If $0 \notin F'(X_k)$, then a zero (if any) of f in X_k is unique (simple).
4. If $0 \notin F'(X_k)$ for some $k \geq 0$, then the asymptotic rate of convergence to a (simple) zero of f in X_k is quadratic (in the sense that $w(X_{k+1}) \leq c[w(X_k)]^2$, c being a constant).
5. If $0 \notin F'(X_k)$, and x_k is the center of X_k, then at least half of X_k is eliminated in the next step. Thus, convergence is rapid even at the initial iterations when $w(X_k)$ is still large.

Systems of Nonlinear Equations

Now we change to vector notation $x = (x_1, \ldots, x_n)^T$ and $f = (f_1, \ldots, f_n)^T$. We wish to solve the system of equations

$$f(x) = 0 \tag{22}$$

globally, that is, to find and bound all of the solution vectors of Eq. (22) in a given box $X^{(0)}$. For noninterval methods, it is sometimes difficult to find one solution, quite difficult to find all solutions, and most often impossible to know whether all solutions are found. In contrast, it is a straightforward matter to find all solutions in a given box by interval methods in a finite number of iterations, proving automatically, at the same time, that there is no other solution in the initial box.

Various interval Newton methods exist for solving Eq. (22) globally. Similarly to the case of a function of one variable, they all iteratively solve a linear interval approximation of Eq. (22). They differ in the choice of the linearization and the way the linearized equations are solved. Most often, Eq. (22) is linearized in the following way. Let $J(x)$ denote the Jacobian matrix of $f(x)$. Similarly to the scalar case [see Eq. (19)], it can be shown that

$$f(y) = f(x) + J(\xi)(y - x) \tag{23}$$

Let $J(X)$ be the interval extension of $J(x)$ in X. It follows from Eq. (23) that, if y is a zero of f in X, then y is also in the solution set S of the system

$$f(x) + J(X)(y - x) = 0 \tag{24}$$

which is a system of linear interval equations with respect to y (x is fixed and is usually the center of X). Let Y denote an interval solution of Eq. (24), that is, a box containing S. The interval Newton method for solving Eq. (22) is based on the following procedure:

$$X^{(k+1)} = X^{(k)} \cap Y^{(k)}, \quad k \geq 0 \tag{25}$$

where $Y^{(k)}$ is an interval solution of

$$J(X^{(k)})(y - x^{(k)}) = -f(x^{(k)}) \tag{26}$$

with respect to y. Because the linear interval system given by Eq. (26) is to be solved repeatedly (for different boxes $X^{(k)}$), approximate methods are used to solve it (the computation of the optimal solution $\tilde{Y}$ would require an unacceptably large amount of computation). The existing interval Newton methods differ from one another, basically, in the way Eq. (26) is solved. In the earlier versions, the interval Gaussian elimination was used. Since then, many other possibilities have been investigated. Thus, in Hansen's method (5), Eq. (26) is first preconditioned using the inverse of the center of $J(X^{(k)})$. The resulting modified system

$$M(X^{(k)})(y - x^{(k)}) = r(x^{(k)}) \tag{27}$$

is then solved in a Gauss–Seidel way.

The interval Newton method generates a list L of boxes awaiting processing. The iterative process is terminated when the list is empty. Indeed, the procedure defined by Eqs. (25), (26) results in one of the following three outcomes:

1. The sequence $X^{(k)}$ converges to a solution $x^{(s)}$ as k increases. Actually, the iterations are stopped whenever the width of $X^{(k+1)}$ becomes smaller than a constant ϵ_1 (accuracy with respect to x). Now $x^{(s)}$ is approximated by the center x^c of $X^{(k+1)}$. If the list L is not empty, a box is retrieved from L. It is renamed $X^{(0)}$ and Eqs. (25), (26) are resumed with $X^{(0)}$.
2. The sequence $X^{(k)}$ converges to a box X^* whose width is larger than ϵ_1. In practice, the procedure is stopped when the reduction in the volume of two current boxes $X^{(k)}$ and $X^{(k+1)}$ becomes smaller than a constant ϵ_2. In this case, $X^{(k+1)}$ is split along its widest side into two boxes X^L and X^R (left and right). The right box is stored in the list L for further processing. The left box is renamed $X^{(0)}$ and the iterative process continues with $X^{(0)}$.
3. At some k, $Y^{(k)} \cap X^{(k)} = \emptyset$. This is an indication that Eq. (22) has no solution in $X^{(k)}$, and $X^{(k)}$ is discarded (not stored in L). A box is retrieved from L (if L is not empty), and the computation process continues as before.

The above algorithm [presented in detail in (4–6)] preserves all of the remarkable properties of the extended-interval Newton algorithm considered before: global and rapid convergence, guaranteed location of all solutions to Eq. (22) contained in the initial region X^0, computational proof of existence, uniqueness or absence of a solution in $X^{(0)}$.

TOLERANCES OF LINEAR CIRCUITS

Various tolerance problems can be formulated in the class of linear electric circuits depending on the type of the circuit studied (dc or ac circuits, with independent or dependent sources, etc.), the nature of variation of the input parameters (independent or dependent variations) and the number and type of the output variables [see (6,11,12)]. For simplicity, only some basic worst-case tolerance problems are presented here.

dc Circuits

Consider a linear dc (resistive) circuit of uncoupled resistors and independent voltage sources. Let m be the number of branches and $(n' + 1)$ be the number of nodes. One of the nodes (say, the $(n' + 1)$th node) is grounded. The worst-case tolerance analysis problem for this class of circuits is formulated as follows: given the nominal values of the branch resistors and source voltages and their tolerances, find the tolerances on the branch currents and/or the nodal voltages. [Several more general dc tolerance problems (including circuits with dependent parameters) are considered in (6).]

To solve the problem considered here, we first need to set up an appropriate system of linear interval equations. With this in mind, using Kirchhoff's law, we write the following system of real equations in vector form

$$Ay = b \tag{28a}$$

with

$$A = \begin{bmatrix} r & -\alpha \\ -\alpha & 0 \end{bmatrix}, \quad y = \begin{bmatrix} i \\ v \end{bmatrix}, \quad b = \begin{bmatrix} u \\ 0 \end{bmatrix} \tag{28b}$$

where r is a diagonal matrix formed by the branch resistances r_ρ, α is the (reduced) incidence matrix, and i and u are the vectors of the branch currents r_ρ and source voltages u_ρ, respectively. Because the nominal values of r_ρ, u_ρ and their tolerances are given, each input parameter belongs to the respective interval R_ρ or U_ρ, that is,

$$r_\rho \in R_\rho, \quad u_\rho \in U_p \tag{29}$$

We seek the intervals of possible values of all currents and all ungrounded node voltages. Thus, we have $n = m + n'$ output variables and $2m$ input parameters. When the components of r and u vary in the intervals given by Eq. (29), Eq. (28) becomes an interval linear system

$$A^{\mathrm{I}} y = B \tag{30a}$$

with

$$A^{\mathrm{I}} = \begin{bmatrix} R & -\alpha^{\mathrm{T}} \\ -\alpha & 0 \end{bmatrix}, \quad y = \begin{bmatrix} i \\ v \end{bmatrix}, \quad B = \begin{bmatrix} U \\ 0 \end{bmatrix} \tag{30b}$$

where R and U are the interval counterparts of r and u, respectively. It is important to emphasize that all components of R and U are independent intervals. This requirement is crucial because most of the existing interval methods solve only such linear interval systems exactly.

Exact Solution. The exact solution $\tilde{Y}$ of the tolerance problem considered is found by the general Rohn method presented earlier. However, its numerical efficiency is improved substantially, if the interval matrix A^{I} from Eq. (30) is inverse-stable, that is, if $|A^{-1}| > 0$, $\forall\, A \in A^{\mathrm{I}}$ [simple sufficient conditions for establishing the inverse-stability of A^{I} are given in (6,10)]. In this case, the set W from Eq. (18) reduces from 2^n to $2n$ vectors W_i which are determined as follows:

$$W_i = \mathrm{sgn}(A_{\mathrm{c}}^{-1})_i, \qquad i = 1, \ldots n,$$
$$W_i = -\mathrm{sgn}(A_{\mathrm{c}}^{-1})_i, \quad i = n+1, \ldots, 2n$$

where $(A_{\mathrm{c}}^{-1})_i$ is the ith row of A_{c}^{-1}. Thus, $\tilde{Y}$ is found by solving the auxiliary Eq. (17) only $2n$ times.

Example 1. We consider the circuit showed in Fig. 2. Each resistor has one and the same nominal resistance $r_k^{\mathrm{c}} = 100\Omega$, $k = 1, \ldots, m$, and an equal tolerance radius $\Delta_k = w(R_k)/2 = 2\Omega$. The source voltages are $e_1^{\mathrm{c}} = e_2^{\mathrm{c}} = 100\mathrm{V}$, $e_5^{\mathrm{c}} = e_7^{\mathrm{c}} = 10\mathrm{V}$ and

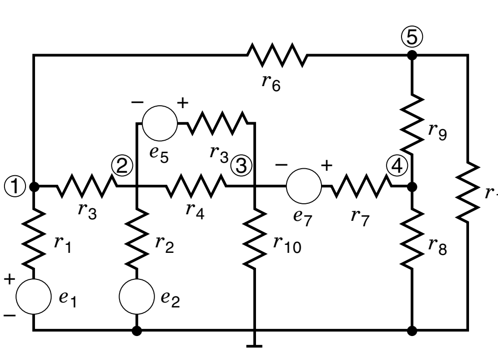

Figure 2. Tolerance analysis of all branch currents and node voltages of the dc circuit shown for a ±2% tolerance on the circuit resistors.

are assumed to have zero tolerances. The problem is to find the intervals of all branch currents i_k, $k = 1, \ldots, m$, and the intervals of all node voltages V_k, $k = m+1, \ldots n$ [the last $(n' + 1)$th node is grounded, i.e., $V_{17} = 0$].

The solution $\tilde{Y}$ is obtained by the simplified version of Rohn's method because the interval matrix A^{I} associated with the circuit studied is inverse-stable. Additionally, the sign-accord algorithm converges every time in one iteration. Thus, the optimal solution $\tilde{Y}$ is found by solving only $2n = 32$ real linear systems of type Eq. (17a). In contrast, the Monte Carlo method currently used in practice require solving Eq. (28) thousands of times to attain the same accuracy.

AC Circuits

In this case, the input parameters x_i additionally include inductances L (mutual inductances M) and capacitances C. We assume that we are interested in one single output variable y and that the relationship $y = f(x)$ between y and the parameter vector $x = (x_1, \ldots, x_n)$ is explicitly known. Typically, y is the rms value of some output voltage or transfer function and $x_i \in X_i^{(0)} = [\alpha_i, \beta_i]$, $x \in X^{(0)} = (X_1^{(0)}, \ldots X_n^{(0)})$. Thus, the worst-case tolerance problem considered is formulated as follows: given the multivariate function $f(x)$ in a given box $X^{(0)}$, find the range $f(X^{(0)})$ of f over the box $X^{(0)}$.

Let $f(X^{(0)}) = [f^{\mathrm{L}}, f^{\mathrm{R}}]$. The endpoint f^{L} is sought as the global solution of the following minimization problem:

$$f^{\mathrm{L}} = \min f(x_1, \ldots x_n), \quad \alpha_i \le x_i \le \beta_i, i = 1, \ldots, n \tag{31a}$$

Similarly

$$f^{\mathrm{R}} = -\min[-f(x_1, \ldots x_n)], \quad \alpha_i \le x_i \le \beta_i, \quad i = 1, \ldots, n \tag{31b}$$

Three interval methods for solving Eqs. (31) have been suggested in (5): the zero-order method (using no derivatives of f), the first-order method, and the second-order method (resorting to first- and second-order derivatives, respectively). They are all based on an algorithm due to Skelboe (13).

Skelboe's algorithm (for bounding f^{L}):

1. Set $X = X^{(0)}$.
2. Bisect X along its widest side into two subboxes X' and X'' of equal width.
3. Evaluate $F^{\mathrm{L}}(X')$ and $F^{\mathrm{L}}(X'')$.
4. Set $b = \min\{F^{\mathrm{L}}(X'), F^{\mathrm{L}}(X'')\}$
5. Enter the subboxes X' and X'' in a list L.
6. Retrieve from L the subbox X^ρ with the lowest $F^{\mathrm{L}}(X^\rho)$, that is, that box for which $F^{\mathrm{L}}(X^\rho) \le F^{\mathrm{L}}(X^\nu)$, $\rho \ne \nu$. Set $X = X^{(\rho)}$ and remove X^ρ from L.
7. If $w(X) > \epsilon$ where ϵ is a prescribed accuracy, return to step 2. Otherwise proceed to the next step.
8. Set $b = F^{\mathrm{L}}(X)$. Terminate.

On exit from the above algorithm, the real number b obtained is a lower bound on F^{L}. If the algorithm is applied to $(-f)$ then, upon termination, $-b$ is an upper bound on f^{R}. We illustrate the ac tolerance problem by the following example.

Example 2. Consider a second-order active RC filter shown in Fig. 3. Its voltage transfer function is given by

$$T(j\omega) = 1/[1 - \omega^2 R_1 R_2 C_3 C_4 + j\omega C_3(R_1 + R_2)]$$

The tolerance on the amplitude $|T(j\omega)|$ was determined for various tolerances on all four parameters of the circuit by an improved first-order method [see (6), sec. 2.4.2]. The numerical evidence shows that the improvement over earlier versions of the first-order method is substantial.

An alternative approach to the ac worst-case tolerance problem is to formulate it as a system of linear equations with complex coefficients. Approximate solutions are thus obtained in (6,11, and 12).

The approach based on global optimization also solves the ac tolerance problem in its probabilistic formulation when the circuit parameters satisfy the Gaussian distribution law [see (6), sec. 2.5]. This problem is computationally more difficult than the worst-case tolerance problem. Nevertheless, numerical evidence shows that the best interval methods are considerably more efficient than the traditional statistical methods as regards computer time requirements.

TRANSIENT TOLERANCE ANALYSIS

Tolerance analysis of transients in linear electric circuits creates a great variety of problems depending on the mathematical descriptions of the transients, on the one hand, the number and nature of the input parameters, and the number of output variables, on the other. Presently, three basic approaches to formulating (and solving) transient tolerance analysis problems are known [see (6)].

Explicit Form Formulation

In this case, there is only one output variable which is some transient current or voltage in the circuit studied. The input parameters are component values, amplitudes of dc or ac excitations and values of initial conditions. The relationship between the input parameters and the output variable must be available in a closed explicit form. Obviously, this is possible only for circuits of a low order of complexity.

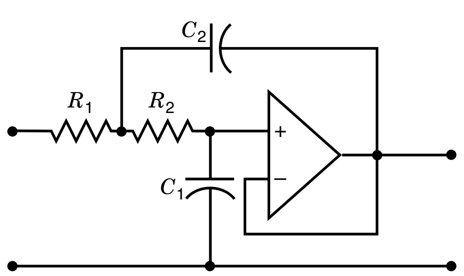

Figure 3. Worse-case tolerance analysis of the voltage transfer function amplitude of a low-pass active filter for various tolerances on the circuit elements.

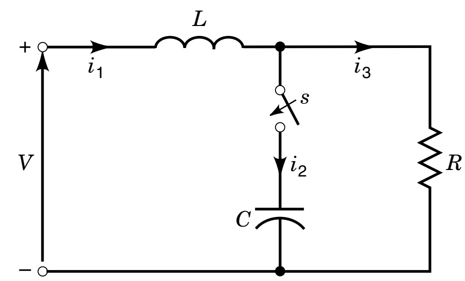

Figure 4. Determination of the dynamic tolerance $I_3(t)$ on branch current $i_3(t)$ for given tolerances on R, L, C, and v.

Example 3. The circuit studied is shown in Fig. 4 [the supply v is constant and $v_c(0) = 0$]. The dynamic tolerance analysis problem considered is to find for fixed (but arbitrary) time t the interval $I_3(t)$ of all possible values of the branch current $i_3(t)$ when L, C, R, and v belong to some prescribed intervals L^I, C^I, R^I, and v^I. We assume that the quantity

$$\delta = \frac{1}{R^2C^2} - \frac{1}{LC} \tag{32}$$

is positive for all $R \in R^I$, $L \in L^I$, and $C \in C^I$. Then the solution $i_3(t)$ is given by the formula

$$i_3(t) = \frac{v}{R}\left[1 + \frac{1}{2CR\sqrt{\delta}}(e^{k_1 t} - e^{k_2 t})\right]$$

where δ is defined by Eq. (32) and k_1, k_2 are given by $k_{1,2} = -1/(2RC) \pm \sqrt{\delta}$. Let $p = (R, L, C, V)$ and $P = (R^I, L^I, C^I, V^I)$. The interval $I_3(t)$ is determined by the range of $i_3(t) = f(t, p)$ when $p \in P$. This is done by using some of the methods for ac tolerance analysis with global optimization.

Based on this example, it is straightforward to present the explicit formulation of the transient analysis of circuits with interval data. Let $p = (p_1, . . ., p_n)$ denote the parameter vector which determines the (scalar) transient $x(t, p)$, and let $p \in P = (P_1, . . ., P_n)$. We assume that the circuit is stable for all $p \in$ P. This assumption is verified by interval analysis methods [see (6), Chap. 4]. Then the set of time functions

$$X(t) = \{x(t, p):\ p \in P, \qquad t \in [0, \infty)\}$$

is called the interval transient because $X(t)$ is an interval for each fixed t. In practice, $X(t)$ is determined for a series of discrete times t_k.

Frequency-Domain Formulation

This is an alternate explicit form for a special dynamic tolerance problem when $x(t, p)$ is the response to a step excitation and the circuit has zero initial conditions. In this case [Kolev (6)]

$$x(t, p) = \frac{2}{\pi}\int_0^\infty \frac{r(\omega, p)}{\omega} \sin\omega t \, d\omega \tag{33}$$

where $r(\omega, p)$ is the real part of frequency response $F(j\omega, p)$ of the circuit investigated for a fixed parametric vector p. Because Eq. (33) expresses the relationship between the output variable $x(t, p)$ and the input parametric vector p in explicit form, the interval solution $X(t)$ is determined as the range of $x(t, p)$ over P for each t. In practice, the integration in Eq. (33) is approximated by a sum applying (say) Simpson's integration rule. Three illustrative examples with circuits containing up to four interval parameters are thus solved in (6, Examples 5.2–5.4).

Time-Domain Formulation

In this formulation, the transients are described implicitly by a system of differential equation (in vector form)

$$\dot{x} = Ax + b(t), \quad t \in [0, \tau], \quad \tau < \infty$$

with initial conditions $x(0) = c$. In the most general case, the elements a_{ij} of A, b_i of b and c_i of c all depend on the input parameter vector p. Thus

$$\begin{aligned} \dot{x} &= A(p)x + \varphi(t)b(p), \quad t \in [0, \tau] \\ x(0) &= c(p) \end{aligned} \tag{34}$$

where $a_{ij}(p)$, $b_i(p)$ and $c_i(p)$, $i, j = 1, \ldots, n$, are generally nonlinear functions of p. Therefore, the solution $x(t, p)$ of Eq. (34), which is now a vector, also depends on p. Once again, we assume that the circuit is stable for all possible $p \in P$. The tolerance problem is to determine the solution vector $X(t) = [X_1(t), \ldots, X_n(t)]$ which corresponds to $x(t, p)$ when $p \in P$. This problem is extremely difficult to solve. Therefore, it is simplified in practice by assuming that a_{ij}, b_i, and c_i are independent and lie in some intervals a_{ij}^{I}, b_i^{I}, c_i^{I} (these intervals are in fact some extensions or the ranges of $a_{ij}(p)$, $b_i(p)$, and $c_i(p)$, respectively, in P). Numerical examples with n varying from 2 to 5 are given in (6, Examples 5.5–5.9).

GLOBAL ANALYSIS OF NONLINEAR dc CIRCUITS

We consider the problem of finding all dc operating points (global dc analysis problem) of nonlinear electric circuits for the case where the nonlinear elements are modeled by continuously differentiable functions.

General Form Description

In this case, the nonlinear dc circuit is described by the vector equation

$$f(x) = 0 \tag{35a}$$

The components x_i of x (branch currents, branch or nodal voltages) are bounded in practice within some admissible intervals, that is $x_i \in X_i^{(0)}$, $i = 1, \ldots, n$ or in vector notation

$$x \in X^{(0)} \tag{35b}$$

where $X^{(0)}$ is an initial box with components $X_i^{(0)}$. The global dc analysis problem is formulated as follows: given the vector CD function f and the initial box $X^{(0)}$, find all the real solutions of Eq. (35).

So far, three versions of the interval Newton method have been used for global dc analysis. The first version (denoted M1) appeals to Hansen's method [given by Eqs. (25) and (27)]. The second version M2 implements Krawczyk's method [set forth in (4–6)]. In this method the interval vector $Y^{(k)}$ participating in Eq. (25) is computed as follows:

$$Y = b(x) + x + [E - J(X)](X - x) \tag{36}$$

where E is the identity matrix. In the third version M3 suggested by Alefeld and Herzberger (4), the Jacobian matrix $J(X)$ is represented as the sum of two matrices as follows:

$$J(X) = D(X) - B(X)$$

where D is formed by the diagonal elements of J, and B includes the remaining elements (with changed sign). Then the vector $Y^{(k)}$ involved in the iterative process Eq. (25) is computed as

$$Y = x - D^{-1}(X)[B(X)(x - y) + f(x)] \tag{37}$$

As seen from Eqs. (36) and (37), the last two methods circumvent the necessity of solving the linear interval Eq. (26) and are therefore computationally more efficient than method M1.

Hybrid Form Representation

It is assumed that the circuit investigated allows the so-called hybrid representation [Chua and Lin (8)], that is,

$$\varphi(x) - Hx - s = 0 \tag{38}$$

where H and s are constant matrix and vector, respectively, and $\varphi_i(x) = \varphi_i(x_i)$, $i = 1, \ldots, n$. Equation (38) could be solved by the general methods mentioned previously. Their computational efficiency, however, is limited to circuits of low dimension n. Indeed, they involve recursive splitting of the initial box $X^{(0)}$ into subboxes $X^{(v)}$, and the number of $X^{(v)}$, and hence the computational effort needed to locate all real solutions of Eq. (38) in $X^{(0)}$ grows exponentially with n. On the other hand, the specific form of Eq. (38) permits elaborating two specialized, more efficient interval methods (referred to as M4 and M5) for global analysis applicable to nonlinear dc circuits of larger size [see (6, sec 6.1.2)]. Method M4 is based on the following iterative procedure:

$$\begin{aligned} Y^{(k)} &= \varphi^{-1}[X^{(k)}] \cap L[X^{(k)}] \\ X^{(k+1)} &= Y^{(k)} \cap \{H^{-1}[\varphi(Y^{(k)})] - s], \qquad k \geq 0 \end{aligned}$$

where $L(X) = HX + \mathrm{s}$. The fifth method M5 is a modification of method M3 which takes into account that now

$$D(X) = \mathrm{diag}\{\varphi_i'(X_i) - h_{ii}, \quad i = 1, \ldots, n\}$$

whereas $B(X)$ is a constant matrix $B = \{h_{ij}, j \neq i, i, j = 1, \ldots, n\}$. Thus, the iterative process defined by Eq. (25) takes on the form

$$Y_i^{(k)} = x_i^{(k)} - \left[\varphi_i(x_i) - h_{ii}x_i - s_i - \sum_{j=1}^{i-1} h_{ij}X_j^{(k+1)} - \sum_{j=i+1}^{n} h_{ij}X_j^{(k)}\right] \Big/ D(X_i^{(k)}) \tag{39a}$$

$$X_i^{(k+1)} = X_i^{(k)} \cap Y_i^{(k)}, \quad k \geq 0 \tag{39b}$$

Furthermore, in the previous methods x is the center of the current box X. In method M5, two new points x_i' and x_i'' (other than x_i) are computed and used in Eq. (39a) at every iteration which improves additionally the numerical efficiency of the method.

Using the five methods presented, numerous examples are solved in (6, sec. 5.1.3) with n changing from 2 to 4. The numerical evidence shows that Method M5 has the best performance characteristics. To illustrate its efficiency, consider the following example:

Example 4. The circuit investigated contains four transistors and is described by the vector Eq. (38) with $\varphi_i(x_i) = 10^{-9}(e^{40x_i} - 1)$, $i = 1, 2, 3, 4$. The circuit has nine operating points. Using Method M5, they are all found within accuracy $\epsilon = 0.01$ after $N = 79$ iterations (ϵ is the width of each solution box containing an operating point). For comparison, $N = 207$ and $N = 143$ for Methods M2 and M4, respectively.

ALTERNATIVE APPLICATIONS

The scope of the interval approach would be incomplete if we do not include the so-called robust stability problem and some aspects of the global analysis of dynamic nonlinear circuits.

ROBUST STABILITY

In the field of electrical, electronics and control engineering, it is of paramount importance to guarantee the stability of the circuit investigated (whatever its functions) even in the presence of some uncertainties about the values of various component parameters. Two basic approaches are known for assessing the robust stability: (1) stability of polynomials with interval parameters and (2) stability of interval matrices.

A famous theorem due to Kharitonov (14) establishes the robust stability of polynomials in the simplest case where the polynomial coefficients are independent intervals. Several attempts to extend Kharitonov's approach to more general stability problems have been made in recent years. In (15), Kharitonov's theorem is generalized to polynomials which have all their zeros in a given sector of the complex plane. A second extension which guarantees that the corresponding dynamic system has only aperiodic behavior is obtained in (15). In a more realistic formulation, the polynomial coefficients are nonlinear functions of a certain number of physical parameters. The problem of assessing the robust stability or certain stability margin in this case is equated to a corresponding global minimization problem [see Kolev (6) and the references cited there]. Interval methods are vastly superior to their point counterparts in solving the latter problem.

Interesting results have also been obtained for the case where the robust stability of the system studied is assessed by the stability of an associated interval matrix [see Kolev (6) and the references cited there]. The stability criteria suggested by Kolev (6) are simple and easy to implement on a computer.

Finally, the interval extension of the Nyquist criterion in (6) is a most effective means for robust stability analysis of feedback circuits or systems. Indeed, the assessment of the gain or phase margin of stability of the closed-loop system is reduced to several ac tolerance analysis problems related to the open-loop transfer function.

Dynamic Nonlinear Circuits

The interval approach has been applied to solve the following problem from global analysis of nonlinear dynamic circuits: given the system

$$\dot{x} = \psi(x, t) \tag{40}$$

where ψ is a T-periodic function in t, we seek all the T-periodic solutions of Eq. (40) when the initial conditions vector x^0 belongs to a box $X^{(0)}$. An interval method for solving the problem, suggested by Kolev (6), is based on an equivalent transformation of the original problem to that of finding all fixed points of the system $x^0 = f(x^0)$ in $X^{(0)}$. The latter is solved by an interval method of zero order. In its present implementation, the method is rather time-consuming and is applicable only to circuits of low dimension ($n \leq 3$).

The challenging problem of establishing the uniqueness of a T-periodic steady state in nonlinear electric circuits has also been considered. A new result has been obtained for the special case where the function ψ from Eq. (40) is of separable form, that is $\dot{x} = \varphi(x) + b(t)$. It has the form of a sufficient condition: the T-periodic solution is unique if an associated interval matrix is stable. The latter problem is handled by some of the methods for assessing robust stability.

PERFORMANCE CHARACTERISTICS

Interval methods have proved reliable for solving numerous problems arising in electrical and electronics engineering. Some of these problems (such as global analysis of systems of nonlinear equations, global optimization) which, in their most general form, were previously intractable, are now routine practice. A convincing example is the global analysis of dc nonlinear circuits. Even in the simple case of resistive circuits containing only one-port nonlinear elements, traditional methods provide misleading conclusions concerning the total number of dc operating points in the circuit studied. Thus, Yamamura (16) suggests a method for finding all solutions of piecewise-linear (PWL) resistive circuits and applies it for global analysis of resistive circuits whose nonlinear elements characteristics are described by continuously differentiable functions. He illustrates his approach by several examples. Example 3 deals with a circuit containing 10 tunnel diodes described by a system of 10 nonlinear equations. Each tunnel diode characteristic is approximated fairly well by 10 linear segments. Yamamura's method locates seven dc operating points in a given box $X^{(0)}$. Application of an interval method [Kolev and Mladenov (17)] shows that this result is incorrect: the total number of operating points for the same circuit and the same box $X^{(0)}$ has been computationally proved equal to nine and all operating points have been located within an accuracy of 10^{-4}. In another example from (16) (Example 4 dealing with the Hopfield neural network that comes from a layout problem of printed boards), the number of solutons changes from 15 to 19 when the number of the approximating linear segments is increased from 30 to 100. In contrast, interval methods never "go wrong."

For most problems solved to date, interval methods require reasonable amounts of computer time which are usually smaller than those needed by traditional methods [see 6,11,12). Theoretically, the complexity of some interval methods, and hence computer time required, may grow exponentially with the dimension n of the problem and the size of the initial box where solutions are sought. This is the case of the exact Rohn's method for dc tolerance analysis, tolerance analysis methods based on global optimization, and global analysis of nonlinear circuits. Judging from the available experimental evidence this difficulty has, however, not occurred in practice.

It should also be borne in mind that some interval methods for global solution of nonlinear problems of higher dimension require relatively larger memory volumes. Typically, this occurs in the case of zero-order methods (using no derivatives of the functions involved), such as the original Skelboe algorithm. At the earlier iterations of the computation process, the current box X (starting with the initial box $X^{(0)}$) is, most often, split into two halves. Each half is, subsequently, subdivided again which generates a long list L of subboxes awaiting processing. If the dimension n of the problem considered and the size of $X^{(0)}$) are large enough, the storage of L requires a bigger memory volume. However, for the type of problems tackled so far, this has not caused any difficulties.

Another peculiarity of the interval methods for global nonlinear analysis (and global minimization) is the so-called cluster effect [e.g., see (18)]. Ideally, for each point solution x^s, an interval method should provide one single interval solution, that is, a small box X^s of width ϵ (ϵ being the solution accuracy) which covers x^s. However, if clustering occurs, on termination of the computation, there are a certain number of small boxes X^{s_1}, X^{s_2}, . . ., X^{s_k} (of the same width ϵ) around each X^s. Unlike X^s, these boxes do not contain the solution x^s and should not be identified as interval solutions. Clustering appears, essentially, because the method does not delete small enough boxes around a solution. The cluster effect grows stronger as the problem dimension n increases. On the other hand, for one and the same problem, the cluster effect decreases if the method used converges better toward a solution. Therefore, clustering is reduced or even completely avoided if higher order methods are employed. This has been observed experimentally with the known interval methods for global dc analysis when n exceeds some "critical" value (which, depending on the problem solved and the method used, varies from $n = 4$ to around $n = 10$). In the present implementation of the nonlinear analysis methods, clustering is detected "manually" by inspection of all point solutions x^s and exclusion of those caused by the cluster effect.

IMPROVED NUMERICAL EFFICIENCY

The drawbacks of most interval methods are caused, essentially, by overestimation. The drawbacks are reduced to a great extent or completely overcome if improved interval extensions are used. Two such techniques aiming at obtaining interval extensions of smaller excess are mentioned here.

Modified Mean-Value Forms

The mean-value form given by Eq. (7) can be modified in a number of ways to ensure a narrower extension. The first improvement is the so-called monotonicity test form [Moore (1)]

$$F_{\mathrm{MT}}(X) = [f(u), f(v)] + \sum_{i \in \mathrm{S}} F_i'(X)(X_i - m_i) \tag{41}$$

where S is the set of integers i such that $F_i'(X)$ properly contains zero and $u_i = x_i^{\mathrm{L}}$, $v_i = x_i^{\mathrm{R}}$, if $F_i'(X) \geq 0$, $u_i = x_i^{\mathrm{R}}$, $v_i = x_i^{\mathrm{L}}$ if $F_i'(X) \leq 0$ and $u_i = v_i = m_i$ if $i \in S$. In Eq. (7) and Eq. (41), the interval extensions $F_i'(X) = F_i'(X)$, . . . X_n) depend, generally, on all of the intervals X_i. Hansen (5) has introduced an improvement in which part of the arguments become real numbers. Because of inclusion monotonicity, this leads to narrower $F_i'(X)$ and, hence, to narrower extensions. Further improvements (introduction of "lower and upper" poles, sequential evaluation of the derivatives, choice of the bisection direction, etc.) are suggested by Kolev (6), secs. 2.2 and 2.4). In accordance with the theoretical predictions, the numerical evidence shows that the best mean-value forms lead to methods of enhanced efficiency.

Interval Slopes

Interval slopes were introduced in interval computations by Krawczyk and Neumaier (19) for rational functions of a single variable. Computation of interval slopes (in fact, of interval extensions of the slopes) in the case of multivariate functions is based on the so-called slope arithmetic [see (20)]. The extension of interval slopes to irrational functions is suggested by Kolev (21). This permits a more efficient analysis of nonlinear dc circuits containing transistors (and diodes) if the Ebers–Moll model of transistors is used.

The slope of a function f of one variable x at x, z is defined as

$$f[x, z] = \begin{cases} [f(x) - f(z)]/(x - z), & x \neq z \\ f'(x), & x = z \end{cases} \tag{42}$$

The interval slope $S(X)$ is either some interval extension $F[X, z]$ or (better) the range $f[X, z]$ of $f(x, z)$ with respect to x in X. Let $D(X)$ denote the interval derivative of f in X. The following important property is then valid:

$$S(X) \subseteq D(X) \tag{43}$$

and the inclusion is usually proper. The above inclusion motivates the use of interval slopes rather than interval derivatives in all of the interval methods for nonlinear circuit analysis. The improved numerical efficiency of the approach based on interval slopes is established experimentally in (17,19–21) where nonlinear systems involving up to 10 equations with nine solutions are solved.

TRENDS FOR FUTURE DEVELOPMENT

Interval approach is expected to develop in two directions. First, it is expected that the scope of the interval approach will be enlarged. So far, interval approach covers basic problems relative to: (1) robustness analysis of linear circuits and (2) global analysis of nonlinear circuits. Many other applications are, however, conceivable in the domain of circuit analysis, such as tolerance analysis of nonlinear circuits, robust stability of certain types of nonlinear systems (for example, Lure's systems), and analysis of chaos in nonlinear circuits.

It is expected that interval approach will also be applied to solve problems in circuit synthesis [such an example is considered in (6), p. 276]. It has already been employed (though in a rather restricted manner to ensure high accuracy of computation when all input data are exact) for the designing of control systems (22).

Some of the interval methods suggested for solving electric circuit analysis problems exploit the specific structure of the class of circuits considered to devise algorithms of improved numerical efficiency. On the other hand, many of the electric circuit analysis problems—tolerance analysis via global optimization, robust stability and performance analysis, and global nonlinear analysis in the case of equations of arbitrary form—are more general. The methods developed for solving these latter problems can, therefore, be applied (directly or after minor modifications) to tackle similar problems arising in systems of arbitrary physical constituency. For this reason, it is hoped that these more general interval methods will also be useful to control engineers, mechanical engineers, system analysts and other specialists striving to employ modern computer methods in their research or application areas.

The second expected development is improvement of the interval method's efficiency. The existing interval methods for circuit analysis are based on the present state of the art of those interval analysis techniques that are related to the topics considered. However, interval analysis is presently undergoing a period of rapid development, and new improved mathematical tools (methods for obtaining narrower interval extensions, for solving more efficiently linear and nonlinear equations, global optimization problems, linear and nonlinear differential equations, etc.) are constantly emerging. Incorporating such new techniques into the body of interval methods for solving electrical and electronics engineering problems is expected to lead to substantial improvement in the numerically efficiency of thse methods. Progress in their software and hardware implementation (implementation of interval methods by parallel computation techniques, development of user-friendly versions, hardware realization of machine interval arithmetic, etc.) will help disseminate the interval methods among more and more specialists in various fields of science and engineering.

BIBLIOGRAPHY

1. T. Sunaga, Theory of an interval algebra and its application to numerical analysis, RAAG Memoirs II, Tokyo: Gakujutsu Bunken Fukyu-kai, 1958.
2. R. E. Moore, *Interval Analysis,* Englewood Cliffs, NJ: Prentice-Hall, 1966.
3. R. E. Moore, *Methods and Applications of Interval Analysis,* Philadelphia: SIAM, 1979.
4. G. Alefeld and J. Herzberger, *Introduction to Interval Computation,* New York: Academic, 1983.
5. E. Hansen, *Global Optimization Using Interval Analysis,* New York: Marcel Dekker, 1993.
6. L. V. Kolev, *Interval Methods for Circuit Analysis,* Singapore: World Scientific, 1993.
7. S. Skelboe, True worst-case analysis of linear electrical circuits by interval arithmetic, *IEEE Trans. Circuits Syst.,* **26**: 874–879, 1979.
8. K. Madsen and H. Jacobsen, Algorithm for worst-case tolerance optimization, *IEEE Trans. Circuits Syst.,* **26**: 775–783, 1979.
9. L. O. Chua and P. M. Lin, *Computer Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques,* Englewood Cliffs, NJ: Prentice-Hall, 1975.
10. J. Rohn, Solving interval linear systems, *Freiburger Intervall-Berichte* **7**: 1, 1984.
11. K. Okumura and S. Higashino, An interval method for worst-case analysis of ac linear networks, *Proc. Int. Symp. on Electr. Eng., ISTET'95,* Thessaloniiesis, Greece, 1995, pp. 446–449.
12. W. Tian and X. T. Ling, Novel methods for circuit worst-case tolerance analysis, *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.,* **43**: 272–278, 1996.
13. S. Skelboe, Computation of rational interval functions, *BIT,* **14**: 87–95, 1974.
14. V. L. Kharitonov, On a generalization of a stability criterion, *Akad. Nauk. Kazakhsoi SSR Izv. Ser. Phys.-Mat.,* **1**: 53–57, 1978.
15. C. B. Soh and C. S. Berger, On the stability properties of polynomials with perturbed coefficients, *IEEE Trans. Autom. Control,* **33**: 351, 1988.
16. K. Yamamura, Finding all solutions of piecewise-linear resistive circuits using simple sign tests, *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.,* **40** (8): 546–550, 1993.
17. L. Kolev and V. Mladenov, Use of interval slopes in implementing an interval method for global nonlinear dc circuit analysis, *Intern. J. Circ. Theory and Appl.,* **25**: 37–42, 1997.
18. A. P. Morgan and V. Sapiro, Box bisection for solving second-degree systems and the problem of clustering, *ACM Trans. Math. Software,* **13**: 152–167, 1987.
19. R. Krawczyk and A. Neumaier, Interval slopes for rational functions and associated centred forms, *SIAM J. Number. Anal.* **22**: 604–616, 1985.
20. S. Zuhe and M. A. Wolfe, On interval enclosures using slope arithmetic, *Appl. Math. Comput.,* **39**: 89–105, 1990.
21. L. V. Kolev, Use of interval slopes for the irrational part of factorable functions, *Reliable Computing,* **3**: 83–93, 1997.
22. G. Ludyk, *Computer-Aided Design of Dynamic Systems,* Berlin: Springer-Verlag, 1990.

LUBOMIR V. KOLEV
Technical University of Sofia

INTERVAL MEASUREMENT. See TIME MEASUREMENT.

INVERTERS. See DC-AC POWER CONVERTERS (INVERTERS).

INVERTERS, NEUTRAL POINT CLAMPED. See BATTERY STORAGE PLANTS.

INVERTERS, STEPPED-WAVE. See BATTERY STORAGE PLANTS.