11

Symbolic Methods in Semiconductor Parameter Extraction

A. Konczykowska
FRANCE TELECOM/ CNET-PAB
Bagneux Cedex, France

W. M. Zuberek
Memorial University of Newfoundland
St. John's, Canada

11.1. INTRODUCTION

Due to rapid developments in semiconductor technologies, which have resulted in increased circuit complexity and improved circuit performance, verifying designs through simulation has become an indispensable part of the integrated circuit (IC) design process. As complex mixed analog/digital circuits are becoming quite popular, and digital circuits use frequency rates for which analog effects must be taken into account during the design process, there is constant demand for more efficient analysis methods for analog as well as digital circuits.

Computer-aided circuit analysis (or circuit simulation) cannot provide reliable results without adequate specification of circuit elements and device models. Existing device models use large sets of parameters, values of which must be determined very carefully to represent device characteristics accurately. Because of highly nonlinear device models, these parameters usually cannot be determined by direct measurements; popular extraction methods use iterative techniques to minimize differences between measurement data and model behavior in the full range of operating conditions.

Iterative extraction of model parameters can be regarded as an optimization process, which minimizes the differences between a set of measurement data and the corresponding circuit responses by adjusting the values of model parameters (used as optimization variables). These differences are usually expressed by one of a variety of error functions. The result of this optimization determines such a set of model parameters, for which the circuit responses are "as close as possible" to the measurement data (in the sense of the error functions used in optimization).

One of the flexible approaches to parameter extraction is to use a circuit simulator rather than a set of model equations (such an approach is called *simulation-based parameter extraction* [1]). An important advantage of the simulation-based method is that the extractor can use all capabilities of the circuit simulator, so all packaging and mounting parasitics can easily be taken into account during the extraction process. Moreover, the extraction can use many different types of measurement data (dc, ac, noise, distortion, etc.) for the best selection of parameter values. On the other hand, repeated simulations can easily become rather time-consuming, especially when numerous parameters are extracted from large sets of measurement data. For linear, frequency-domain analyses, the dependence of circuit responses on some variables can be derived in a symbolic form. This symbolic form can then be used very efficiently for finding circuit responses of the same circuit for different combinations of parameter values. Many time-consuming computations can be eliminated using such an approach.

This chapter briefly describes a simulation-based parameter extraction program and then indicates where symbolic simulation can successfully replace the traditional numerical approach. An interface between symbolic and numerical simulators is outlined and the integration of these two approaches is discussed in the context of parameter extraction. Practical examples of parameter extraction are used as an illustration as well as the basis of some performance comparisons.

11.2. OVERVIEW OF PARAMETER EXTRACTION

11.2.1. Approaches to Parameter Extraction

The goal of parameter extraction is to determine such values of device model parameters that minimize the (total) differences between a set of measured characteristics and results obtained by evaluations of the device model. This minimization process is often called *fitting* of model characteristics to the measurement data. The comparison is performed using some error functions, such as absolute or relative l_p-norm, logarithmic norm, etc., that determine the (total) differences between measurement data and evaluated results. Quite often several different error functions are used for different types of data (such as dc data, ac data, etc.).

Several different approaches to parameter extraction have been proposed. They include general or specialized, and direct or iterative extraction methods. Specialized methods extract some subsets of model parameters, for example, model resistances, or capacitances, or dc parameters only [2]–[4], while general methods determine all parameters of the model. Direct extraction methods approximate model equations by linear functions and determine the values of parameters graphically or by solving linearized equations [4], [5], while iterative methods fit the model responses to a set of measured characteristics by minimizing an objective function that quantitatively characterizes the fit [6]–[9]. Sometimes a mixed approach is used in which some parameters are extracted using the direct methods and the remaining ones by an iterative procedure [2], [10], or the direct solution is used as the starting point to an iterative refinement [11], [12].

Iterative methods can be equation-based or simulation-based. Equation-based methods use a set of model equations to evaluate device responses corresponding to measurement data [6], [13]. In the simulation-based approach, a circuit simulator (or its part that evaluates devices and their models) is used to provide circuit responses. The simulation-based approach eliminates potential inconsistencies between model equations used by the extractor and the equations implemented in simulation tools. It also provides, at the extractor level, many simulation capabilities that are supported by the circuit simulation tool used.

Several extraction programs have successfully demonstrated applicability of general optimization methods to parameter extraction [6]–[9]. However, the convergence properties of these methods depend on properties of the error functions; typically, it is required that the error functions have no singularities, be unimodal, and be approximately quadratic in the region of a minimum. These conditions are not always met by popular error functions [14], especially in the absence of good initial estimates of parameters. To achieve convergence, global optimization methods are used [15], [16]. However, global methods are rather inefficient, so an alternative approach replaces the global optimization by a sequence of "partial" extractions [17], each of which is performed using a subset of parameters and a subset of measurement data. Also, less efficient but robust optimization methods are being used [8] in order to avoid convergence problems of gradient techniques.

Several programs for parameter extraction are available on a commercial basis. Brief characteristics of some of them are as follows.

Hewlett-Packard offers an integrated circuit characterization and analysis program (IC-CAP) [18]. Presented as an open framework for computer-aided engineering (CAE) modeling applications, this tool differs considerably from a previous HP modeling product, TECAP. Standard and user-defined models are supported, as well as different measurement types. Parameter extraction language (PEL) can be used to develop specific extraction tasks.

HarPE, from Optimization Systems Associates Inc. (OSA), is an equation-based extractor dedicated to nonlinear device simulation, characterization, and optimization. Frequency-domain large-signal, small-signal, and dc measurements are used for extraction. HarPE was developed primarily for microwave applications [19].

SUCCESS is a product of ANACAD-EES [20]. The program provides model parameter extraction for various devices (bipolar, MOSFETs, diodes, etc.), as well as for user-defined models. The program is designed to extract parameters in a user-specified constrained region of operation. The extractor is equation-based and uses a modified Levenberg–Marquardt algorithm for optimization.

Universal Transistor MOdelling SofTware (UTMOST) is available from SILVACO [21]. This program determines model parameters from measurement data via direct parameter extraction and/or optimization methods for various types of measurements and a wide range of devices.

11.2.2. Parameter Extraction as an Optimization Problem

Iterative parameter extraction is based on repetitive comparisons of measurements obtained from fabricated devices with the model evaluation results corresponding to different values of (extracted) parameters. The iterative procedure adjusts the values of parameters in a way that minimizes the value of the error function and terminates the extraction when no further improvement can be made.

Parameter extraction can be formulated as an optimization problem in which a nonlinear objective function F is minimized with respect to a set of model parameters P, subject to a set of constraints C. The objective function F describes the (total) differences between the measurement data D and the results of model evaluation $S(P)$:

$$F(D, P) = E(D, S(P)) \tag{11-1}$$

where E denotes a (general) error function.

The optimization process determines such a vector of parameter values P^* that minimizes the objective function

$$F(D, P^*) = \min_{P \text{ subject to } C} (F(D, P)) \tag{11-2}$$

where C determines lower and upper bounds on parameters (e.g., resistances must be positive and not greater than certain physical limits, similarly capacitances, etc.).

The set of data D usually contains different types of measurement data (dc, ac, noise, etc.). Different error functions can be used for different data types. Moreover, in order to speed up the extraction process and to improve its convergence, quite often an extraction strategy is developed that decomposes the extraction process into a sequence of "partial extractions," i.e., a sequence of extractions performed for (small) subsets of extracted parameters and subsets of corresponding measurement data [1]:

$$F(D, P^*) = \min_{i = 1, 2\ldots} \left(\min_{P_i \text{ subject to } C_i} (F_i(D_i, P_i)) \right) \tag{11-3}$$

where $D_i \subseteq D$, $P_i \subseteq P$ and $C_i = C$ restricted to P_i.

Usually only the last steps of fine tuning involve large sets P_i of parameters and large sets D_i of data.

11.3. INTEGRATED PARAMETER EXTRACTION

11.3.1. Simulation-Based Parameter Extractor

An iterative, simulation-based and data-driven extraction program, called FIT [1], has been developed using general optimization methods and an "open" circuit simulator SPICE-PAC [22] rather than a set of explicit model equations. The selection of the simulation-based approach rather than the equation-based one was due to FIT's anticipated use in parameter extraction for new, experimental semiconductor devices. The flexibility of the circuit simulation tool is of primary importance for such applications.

SPICE-PAC is a SPICE-compatible circuit simulator. It provides the same set of circuit analyses as SPICE [23], but also contains a number of extensions that are not available in SPICE programs. The main difference between SPICE (and other SPICE-like simulators) and SPICE-PAC is in their structure. SPICE is a "closed" program with a fixed set of circuit elements and circuit analyses, while SPICE-PAC is an "open" package that can easily be combined with other computer-aided design (CAD) tools, for example, optimization methods, symbolic and statistical simulators, circuit extractors, and so on.

Because FIT is designed in a modular way, many extensions can be implemented in a rather straightforward way. For example, device thermal effects can be taken into account [24] or extraction can be performed with respect to a combination of technological, geometric, and electrical parameters [25]. FIT supports integrated as well as partial extraction performed on arbitrary subsets of measurement data and arbitrary subsets of parameters. It uses two optimization algorithms; the initial optimization is usually performed by the very robust direct search method of Nelder and Mead [26] (also known as *downhill simplex*), while a more efficient gradient-based method (from the NAG library [27]) is used in the neighborhood of the solution.

The organization of data for a simulation-based extractor is partially implied by its circuit simulation tool. FIT uses three different types of input data: the circuit description, the definitions of variables, and the measurement data. Measurement data normally include dc measurements, frequency-domain (ac), and/or time-domain (TR) measurements (used also for large-signal analysis of the periodic steady state); harmonic and noise measurements can also be handled by the extractor. Measurements of the same type (e.g., ac for a given bias point, steady state for a given frequency, etc.) form a data "group." The measurement data are simply organized as a sequence of data groups.

These three types of data (the circuit description, the specification of variables, and the measurement data) are processed by different modules of FIT, as shown in Fig. 11.1. The main components of FIT are

General driver: coordinates all remaining parts of the program and performs interaction with the user using an interpreter of a simple command language that describes consecutive steps of the extraction process.

Variables manager: controls the set of optimization variables. It selects subsets of variables for partial extractions and updates the values of variables (as well as corresponding circuit parameters) after each optimization step.

Data manager: maintains a collection of all measurement data and corresponding simulation results. It selects data for partial extraction and stores the results of circuit simulations.

Optimizer: selects the optimization method and adjusts optimization parameters accordingly; it also selects the starting point for the optimization.

Circuit simulator: performs the analyses required by the evaluation of the objective function, i.e., analyses that correspond to the selected measurement data.

A more detailed description of the FIT program is given in Refs. [1] and [28].

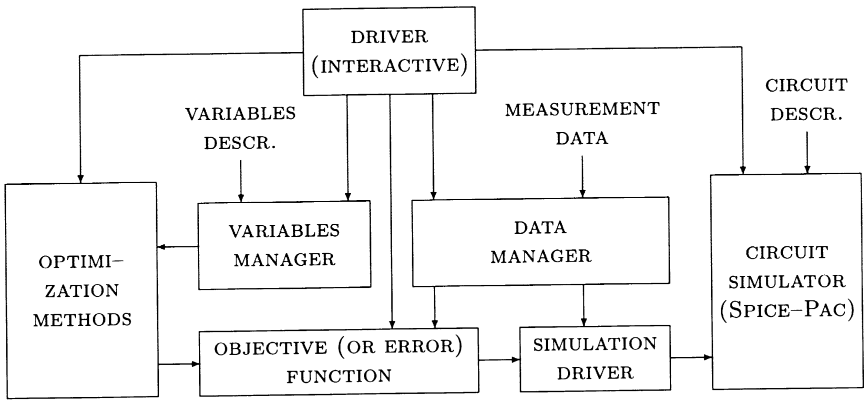


Figure 11.1 General organization of the FIT program.

11.3.2. Numerical Methods in Parameter Extraction

The popular numerical circuit simulators use a modified form of nodal analysis (modified to take care of voltage sources, floating sources, and inductive elements) and Newton–Raphson iteration to solve the system of simultaneous nonlinear algebraic equations [29]

$$F(X) = 0 \tag{11-4}$$

describing the balance of currents at the nodes of the network in terms of node voltages (and some branch currents) X. The solution is typically obtained through a sequence of linear approximations to the nonlinear function $F(X)$ at points $X^{(j)}$

$$G(X^{(j)})\Delta^{(j)} = -F(X^{(j)}) \tag{11-5}$$

where G is the Jacobian of F with respect to X (evaluated at $X^{(j)}$) and the solution $\Delta^{(j)}$ is used to determine the next approximation $X^{(j+1)} = X^{(j)} + \Delta^{(j)}$. The iteration terminates when $\Delta^{(j)}$ is sufficiently small.

This basic scheme is used in the dc operating point, dc transfer curve, and even time-domain analysis; in the latter case, the dependence upon time is eliminated by approximating the differential equations by difference equations (using some numerical integration formula) [30], [31]. Only frequency-domain (small-signal) analyses are significantly different, because they require (for each frequency) a solution of a system of simultaneous linear equations in the complex domain; this is often done by separating the real and imaginary parts of the coefficients and variables, and solving a twice-as-large system of linear equations in the real domain.

A general data-driven and simulation-based optimization scheme (with emphasis on frequency-domain analyses), as implemented in the FIT program, is shown in Fig. 11.2.

11.3.3. Symbolic Methods in Parameter Extraction

For simulation-based parameter extraction in general, but especially in the case of analog (and in particular microwave) applications, a significant part of simulations is performed for the small-signal linear behavior of the circuit. These linear analyses can conveniently be done using symbolic methods rather than numerical ones. The analyzed circuits are typically very small, and the number of symbols can be significantly reduced by eliminating, as soon as possible, all those symbols whose values cannot be modified during analyses.

The principle of symbolic simulation is to derive analytic (or symbolic) functions describing the circuit responses, using symbols representing (some) circuit parameters (rather than their numerical values). Evaluation of these symbolic functions for specific values of symbols provides the required responses of the circuit.

Circuit functions that are used in parameter extraction are driving-point immittances and various matrices (**Y**, **Z**, **A**, **S**, etc.) for a two-port circuit representation. All these circuit functions can be obtained easily from characteristic polynomials of the analyzed circuit [32].

```
continue_optimization := true;
while continue_optimization do
    update_the_values_of_optimization_variables;
    error_value := 0;
    for each data_group do
        if frequency_domain(data_group) then
            update_op_point_voltages_and_currents;
            find_the_operating_point_solution;
            for each frequency do
                    find_the_solution_of_linear_equations(res);
                    error_value := error_value + differences(data,res)
            endfor
        else
            find_circuit_responses(res);
            error_value := error_value + differences(data,res)
        endif
    endfor;
    if end_of_optimization(error_value) then
        continue_optimization := false
    else
        select_new_values_of_parameters(error_value)
    endif
endwhile;
```

Figure 11.2 An outline of a simulation-based data-driven optimization scheme.

For linear, lumped, and stationary (LLS) circuits, the transfer functions of a two-port network are in the form of rational functions of complex frequency *s*:

$$H(s) = \frac{F_j(s)}{F_k(s)} \tag{11-6}$$

in which the numerator F_j and the denominator F_k are characteristic polynomials of the two-port.

Characteristic polynomials F_i can be expressed as

$$F_i(s) = \sum_{l=0}^{n_i} s^l A_{il}(x_1, \ldots, x_m) \tag{11-7}$$

where coefficients $A_{il}(x_1, \ldots, x_m)$ are (nested or expanded) multilinear functions in symbolic elements $x_1, \ldots, x_m$.

In the fully expanded form, the polynomial coefficients are in the sum-of-product form:

$$A(x_1, \ldots, x_m) = \sum_{k=1}^{p} C_k \prod_{l=1}^{r} x_{kl} \tag{11-8}$$

where C_k are real numbers, x_{kl} are circuit symbols, and p and r depend on the topology of the analyzed circuit.

By extracting common factors in (11-7), $F_i(s)$ can be represented as

$$F_i(s) = s^{k_i} T_i \sum_{j=0}^{n_i} s^j R_{ij} \tag{11-9}$$

where each T_i is a product of a constant C_i and some symbols x_{ik}, $k = 1, \ldots, m_i$

$$T_i = C_i \prod_{k=1}^{m_i} x_{ik} \tag{11-10}$$

and each R_{ij}, $j = 0, 1, \ldots, n_i$, is a sum of products

$$R_{ij} = \sum_{k=1}^{l_{ij}} C_{ijk} \prod_{l=1}^{m_{ijk}} x_{ijkl} \tag{11-11}$$

The expanded representation (11-9) is used in a practical implementation of the symbolic analysis. The generated symbolic functions are represented by tables of coefficients C_i and C_{ijk} and the products of symbols $x_1, \ldots, x_n$ [28], [33].

In the context of parameter extraction, and in particular for a sequence of partial extractions, the symbolic functions can be further refined by eliminating all those symbols that cannot change their values during an optimization. Generally, the symbols can be subdivided into two disjoint classes:

fixed symbols, which do not change their values during a single optimization; all such symbols can be eliminated during the generation of symbolic functions by using their numerical values and combining (i.e., multiplying and possibly adding) all such values together, and

variable symbols, which include optimization variables as well as all symbols that depend on the operating point solution (used for the calculation of small-signal parameters).

For each optimization, only the variable symbols are used in the symbolic representation of characteristic functions. In typical partial extractions, there are only two or three optimization variables, so the number of variable symbols is rather small, and the symbolic functions are quite simple.

The generation of symbolic functions follows the approach used in SYBILIN [34], a symbolic simulator developed for analog and, especially, microwave applications.

In general, a different refined form of symbolic functions is needed for each partial extraction. In order to simplify the generation of symbolic functions, the generation is composed of two steps. The first step, performed only once, creates the Coates flowgraph of the circuit with variables corresponding to circuit nodes, as in the modified nodal analysis [35], [36]. This flowgraph is then used to generate all 0-connections, and these 0-connections are used to generate symbolic functions of the circuit. 0-connections [37] are subgraphs of the Coates flowgraph composed of such node-disjoint directed loops that are incident with all graph nodes. The second step, performed for each optimization (as it depends on the set of optimization variables), uses 0-connections to generate symbolic functions with variable symbols only.

For frequency-domain analyses (Fig. 11.2), once the values of variable symbols are determined (by the optimization routine as well as by the solution of the operating point), the symbolic functions can be further reduced by evaluating all the T_i and R_{ij} terms [(11-10), (11-11)]. This results in reduced symbolic functions $\hat{F}_i$:

$$\hat{F}_i(s) = s^{k_i}\hat{T}_i \sum_{j=0}^{n_i} s^j \hat{R}_{ij} \tag{11-12}$$

where all $\hat{T}_i$ and $\hat{R}_{ij}$, $j = 0, 1, \ldots, n_i$, are constants, provided that no frequency-dependent elements are used; they are obtained by the evaluation of T_i and R_{ij} [(11-10), (11-11)].

Only this very simple polynomial form [(11-12)] needs to be evaluated in the innermost (i.e., frequency) loop (Fig. 11.2).

11.3.4. Integration of Numerical and Symbolic Extraction

The addition of symbolic simulation to the extractor FIT resulted in the FIT-S version of the program [28], [33].

Any integration of numerical and symbolic simulation must provide some sort of interaction between these two methods. In the integrated numerical/symbolic parameter extractor FIT-S, the interaction is performed through an interface that supports only a few operations (implemented as interface procedures):

```
RESET (NAME)
NEXTEL(DESC,TYPE,NODES,LEN)
GETVAL(DESC,TYPE,VALUES,LEN)
```

RESET must always be used as the first operation, before any other operation of the interface, because it initializes extraction of circuit elements for symbolic analysis; its parameter NAME must be a subcircuit expansion name (i.e., an X-name in the SPICE convention), which indicates a subcircuit for symbolic analysis.

NEXTEL returns a descriptor DESC, a type TYPE, and a list of nodes NODES of length LEN of the next circuit element (or indicates that the "next" element does not exist); it is implemented in such a way that consecutive invocations of this operation return descriptions of consecutive subcircuit elements (according to the internal representation of the circuit); zero returned as the value of DESC indicates that there are no more elements.

GETVAL uses a vector VALUES to return the numerical values of parameters associated with an element identified by DESC and TYPE; LEN is set to the number of values returned in VALUES.

A typical sequence of interface operations is shown in Fig. 11.3. A symbol table, created in this phase, combines all attributes of all symbols used in the simulation and extraction. These attributes include the class of symbols (fixed, variable) and the values of symbols. The last step shown in Fig. 11.3, **generate_0-connections**, performs the first step of the symbolic analysis.

Integration of the second part of symbolic simulation with the general scheme of parameter extraction (shown in Fig. 11.2) is outlined in Fig. 11.4. The step **generate_symbolic_terms** generates the coefficients T_i and R_{ij} [(11-10), (11-11)] of the symbolic function $F_i(s)$, while **evaluate_coefficients_of_reduced_functions** evaluates $\hat{T}_i$ and $\hat{R}_{ij}$ [(11-12)] using the retrieved values of variable symbols.

The step **retrieve_the_values_of_variable_symbols** is outlined in Fig. 11.5. At each iteration step, once the values of optimization variables are updated and the solution of the operating point is known, the values of all variable symbols are retrieved from the circuit description and stored in the symbol table. It should be noted that the values of all fixed symbols are retrieved from the circuit before this stage (see Fig. 11.3).

```
reset(name);
graph := empty;
nextel(desc,type,nodes,num);
while desc > 0 do
    add_to_flowgraph(symbol,nodes,graph);
    add_to_symbol_table(desc,type,symbol);
    if fixed_symbol(desc,type) then
        getval(desc,type,value,len);
        store_in_symbol_table(symbol,value,len)
    endif;
    nextel(desc,type,nodes,num)
endwhile;
generate_0-connections(graph);
```

Figure 11.3 Initial part of symbolic analysis.

```
continue_optimization := true;
while continue_optimization do
    update_the_values_of_optimization_variables;
    generate_symbolic_terms;
    error_value := 0;
    for each data_group do
        if frequency_domain(data_group) do
            update_op_point_voltages_and_currents;
            find_the_operating_point_solution;
            retrieve_the_values_of_variable_symbols;
            evaluate_coefficients_of_reduced_functions;
            for each frequency do
                evaluate_reduced_functions(val);
                convert_to_circuit_responses(val,res);
                error_value := error_value + differences(data,res)
            endfor
        else
            find_circuit_responses(res);
            error_value := error_value + differences(data,res)
        endif
    endfor;
    if end_of_optimization(error_value) then
        continue_optimization := false
    else
        select_new_values_of_parameters(error_value)
    endif
endwhile;
```

Figure 11.4 An outline of simulation-based optimization with symbolic frequency-domain analysis.

```
for each symbol in symbol_table do
    if variable(symbol) then
        getval(desc,type,value,len);
        store_in_symbol_table(symbol,value,len)
    endif
endfor;
```

Figure 11.5 Implementation of **retrieve_the_values_of_variables_symbols**.

11.4. EXAMPLES

EXAMPLE 11.1—MODELING AND CHARACTERIZATION OF PASSIVE ELEMENTS

Passive elements are normally characterized by relatively simple frequency-domain small-signal models. However, for higher frequencies, the nonlinear effects and frequency-dependent phenomena must be taken into account. Circuit functions in their symbolic form are useful in the selection of the model's topology.

In order to predict the correct behavior of microwave or high-speed circuits, the simulation must take into account models of both passive and parasitic elements. Simulation techniques based on electromagnetic (EM) analysis are not useful in a typical design cycle because of significant memory and computation time requirements. A parametrized library of electrical models was established for the family of passive and parasitic elements such as MIM and interdigitated capacitors, inductors, coplanar capacitors and inductors, bonding wires, TAB (taped automated bonding), bonding pads, vias, etc. [38], [39]. For each element, simplified analytical EM simulation was performed for a wide range of different geometric parameters. Several test designs were fabricated and measured. The topology of small-signal models was investigated using the FIT-S program, and numerical values of model parameters were determined from measurements and compared with those obtained by simulation. Then, for each model parameter, simple analytical formulas in terms of geometric parameters were established. These formulas, obtained using the least square method, can easily be used in circuit simulation in the range of geometries specified for this library.

The following example illustrates the extraction part for a coplanar capacitor. Figure 11.6 shows a small-signal model of a coplanar capacitor from Fig. 11.7. The measurement data (for l = 130 μm, W = 14 μm, and g = 10 μm) as well as the results obtained by simulation are presented in Fig. 11.8.

EXAMPLE 11.2—EXTRACTION OF HBT PARAMETERS

Heterojunction bipolar transistors (HBT) are recognized as promising candidates for high-performance circuits, which include monolithic microwave integrated circuits and high-speed digital circuits. For all such applications, an accurate HBT CAD model is essential for simulating the behavior of high-frequency circuits [12].

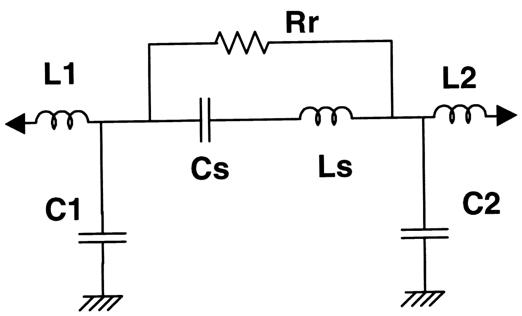


Figure 11.6 Electrical model of a coplanar capacitor.

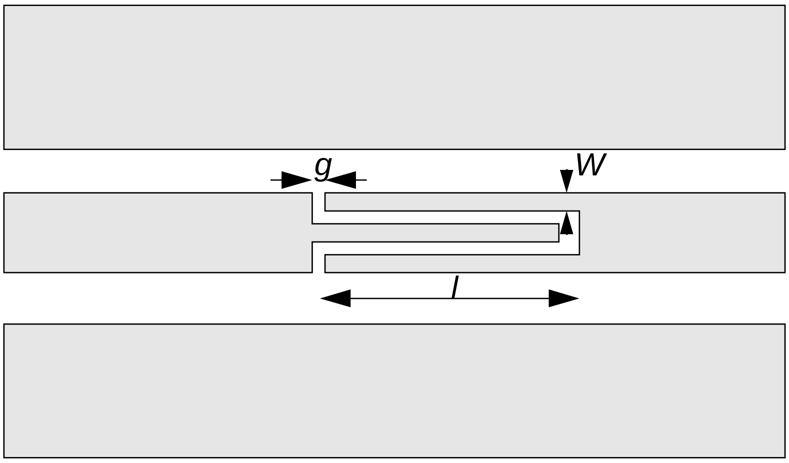


Figure 11.7 A coplanar capacitor layout.

An active device model must accurately represent the static and dynamic behavior of the device. Usually, dc as well as ac measurements are necessary to obtain accurate model parameters. For bipolar transistors, the dc measurements are performed for various configurations and the ac measurements are performed for different bias points in the required range of frequencies.

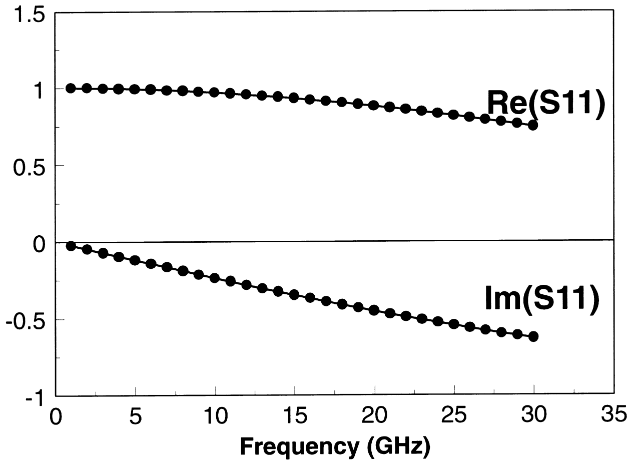
Figure 11.9 shows dc Gummel plot forward characteristics of a GaAs HBT. Families of I_c vs.V_{ce} and V_{be} vs.V_{ce} characteristics, with I_b as parameter, are presented in Fig. 11.10 and Fig. 11.11, respectively. Both families show the characteristic self-heating effect (negative slope). This thermal effect is taken into account during parameter extraction by a combined dc-thermal analysis [24]. Measured and simulated values of *S*-parameters, for one bias point, are presented in Fig. 11.12.

EXAMPLE 11.3—MODEL TUNING FOR NEW TECHNOLOGIES

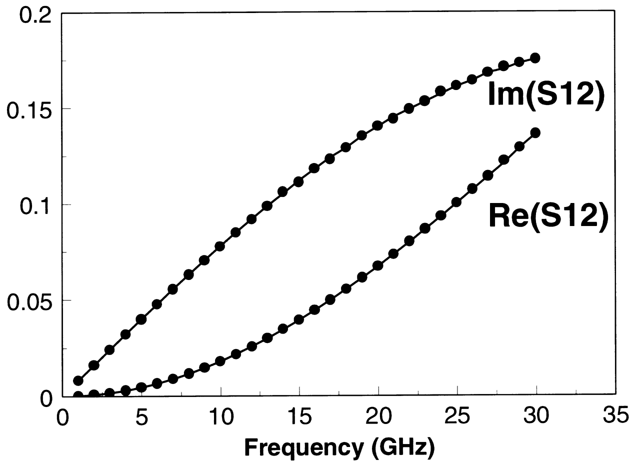
The extraction of parameters is one of the possible techniques used in development of models for new technologies. In this case, the number of ac measurements (for different bias points) is usually rather large (50 to 80) because the character of influence of the bias point on the small-signal parameters is not known precisely, and the whole domain of bias conditions (voltages and/or currents) must be covered by measurements. In the case of known models, the nonlinearities are usually characterized by a small number of parameters, so the extraction can be done with a small number (5 to 10) of properly chosen ac data sets.

Because of the large number of ac data sets, the efficiency of ac simulation is of primary importance for the extraction process. Moreover, special extraction strategies should be established for each new type of device, in order to get reliable, physically meaningful values of parameters, with as little computational effort as possible.

The FIT-S extractor was used in model development of a submicron (0.25 µm) GaAs FET on InP substrate [40]. The topology of the small-signal model is shown in Fig. 11.13. The goal of this extraction was to obtain the characteristics of major small-signal parameters in the function of bias voltages (V_{gs} and V_{ds}).



(a)



(b)

Figure 11.8 Measured (dots) and fitted (line) *S*-parameters of a coplanar capacitor: (a) parameter S_{11}; (b) parameter S_{12}.

The characterization process was organized as follows. The values of small-signal parameters were extracted for each bias point, and were used for parameter characteristics in terms of bias voltages. Nonlinear large-signal formulas for model parameters were then determined and verified using the characteristics obtained earlier.

Figure 11.14 shows two extracted parameters in function of bias voltages.

11.4.1. Evaluation of Examples

The FIT-S extractor was used in all three preceding examples. Integration of numerical and symbolic simulation in the extraction program resulted in significant reduction of the execution time of ac analyses.

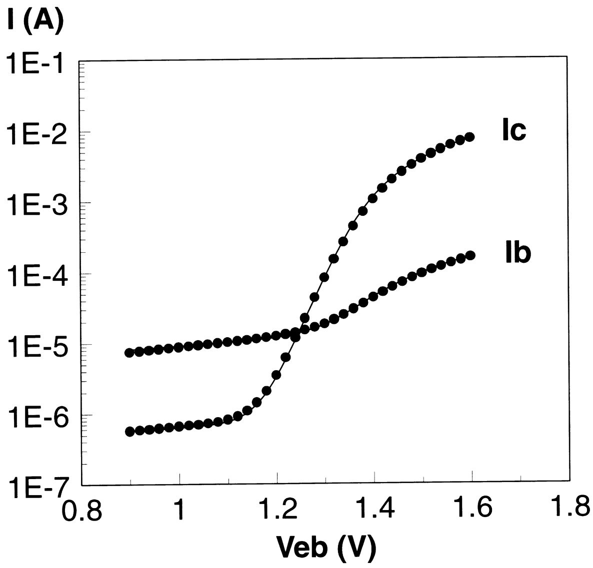


Figure 11.9 Gummel plot dc characteristics of GaAs HBT.

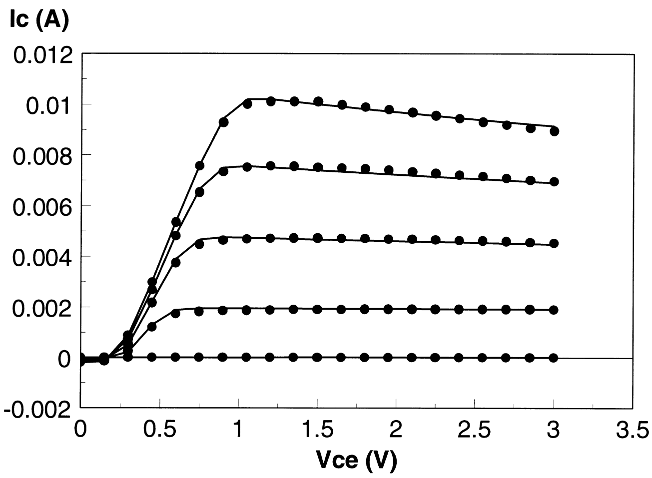


Figure 11.10 I_c-V_{ce} characteristics of GaAs HBT.

The reduction of the computation time due to the introduction of symbolic analysis is especially important for model tuning (Example 11.3). In fact, in this case, the majority of extractions are performed for frequency-domain measurements, with a large number of ac data sets. In the case of extraction of HBT parameters (Example 11.2), the reduction of ac extraction time was also important, but it was the flexibility of the simulation tool that was critical for modeling the self-heating effects.

In the case of development of a CAD library of passive elements (Example 11.1), FIT-S was useful on two fronts: the symbolic form was helpful in the development of

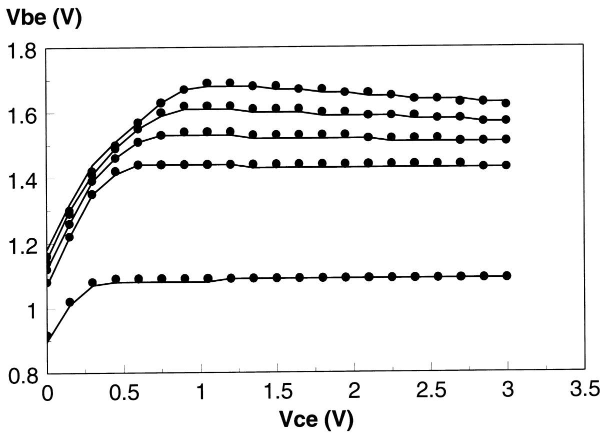


Figure 11.11 V_{be}-V_{ce} characteristics of GaAs HBT.

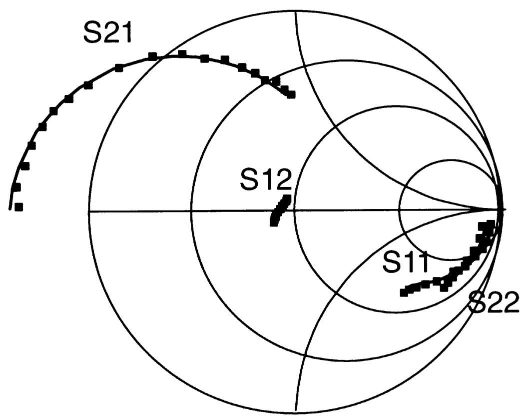


Figure 11.12 *S*-parameters of GaAs HBT.

model topology, and FIT-S was also used in extraction of parameters from frequency-domain measurements, where, as in previous cases, a significant reduction of the execution time was obtained.

In the examples discussed previously, the use of symbolic simulation reduced the execution time of frequency-domain analyses from 13 to 17 times. However, the evaluation of symbolic functions is only a part of all computations performed during parameter extraction. The values of symbolic functions must be converted into *S*-parameters, they must be stored in a database of results, compared with the corresponding measurement values to update the value of the error function, etc. Therefore, a more realistic performance gain is obtained by comparing the total execution time with and without symbolic analysis for a typical extraction process. The result of such a comparison depends on the number of frequency points, and, for previous examples, varied between three and eight times.

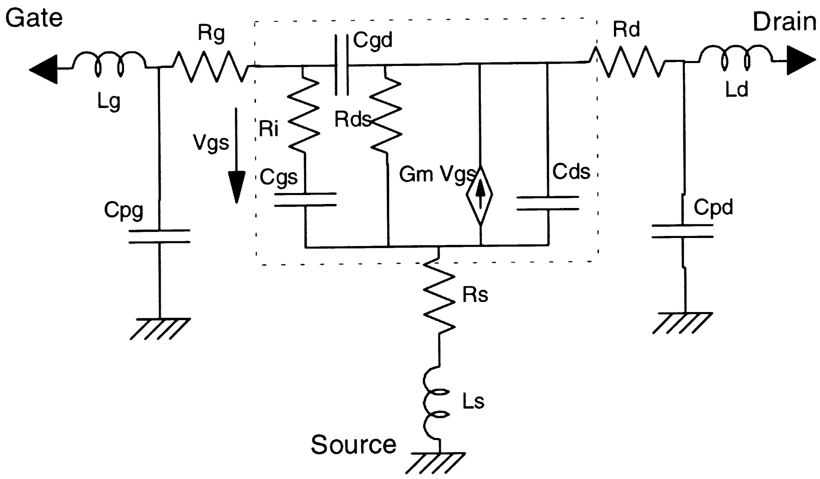


Figure 11.13 Small-signal model for GaAs FET.

11.5. CONCLUDING REMARKS

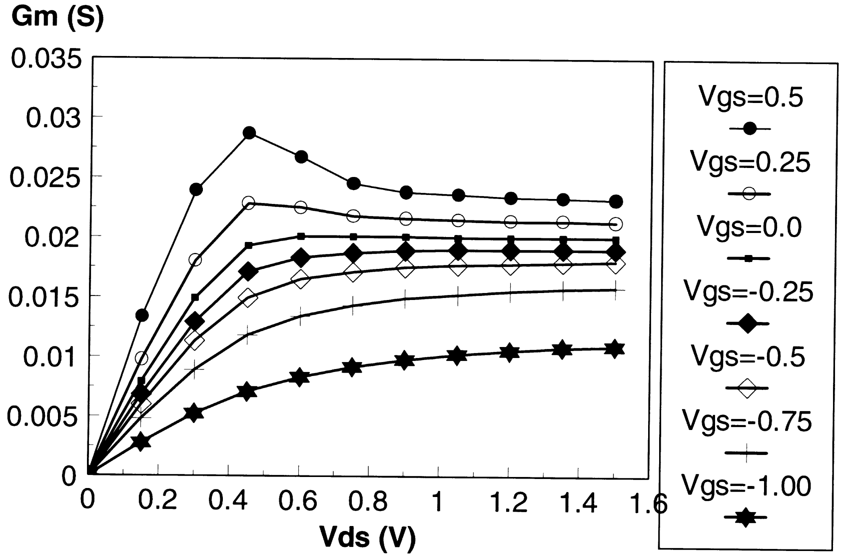
Extraction of model parameters through optimization is a practical approach, especially for models for which direct extraction methods are not practical (thermal effects, parameter conversion, etc.). To obtain parameters of large-signal models, both static and dynamic characterization must be performed.

An integration of symbolic analysis with traditional numerical simulation can significantly reduce the simulation time for frequency-domain analyses. In the case of parameter extraction, this reduction can be used for more sophisticated extraction strategies or optimization methods, which—in general—are more computationally demanding.

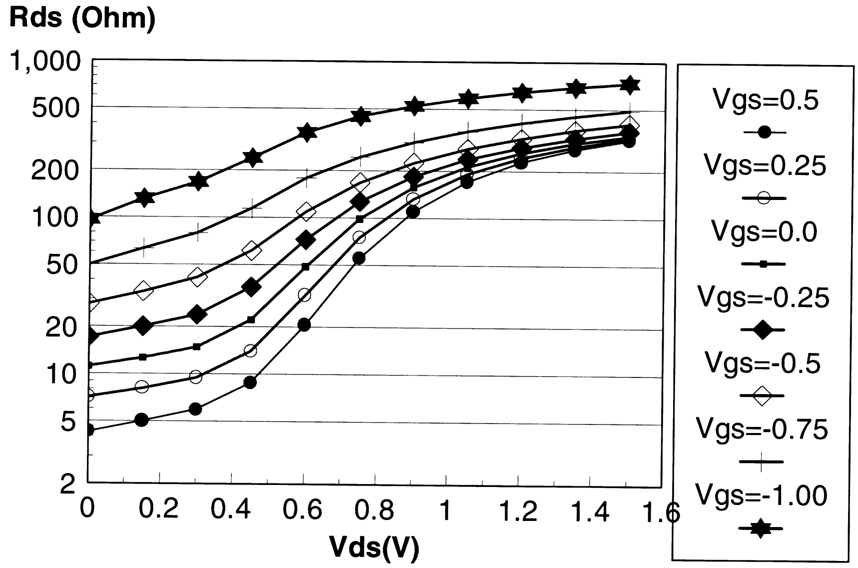
The speed-up of the simulation is not the only advantage of the integrated numerical/symbolic extraction. The symbolic form of (some) results can be useful in model development. Availability of symbolic expressions can help in the search for adequate model structure; it can also be useful in the determination of the most sensitive parameters for different data sets; this, in turn, is helpful in developing efficient extraction strategies. Another possible enhancement in the domain of model development is an automated search of optimal topology for the modeled device (e.g., the algorithms presented in chapter 8).

The selection of the internal representation of symbolic functions should be guided by an efficient evaluation of these functions. The table representation used in FIT-S is only slightly less efficient than a binary code or a compiled high-level code, while it eliminates disadvantages of these other solutions (nonportability, a need of dynamic compiling and linking) [41].

In parameter extraction, the evaluation of symbolic functions is only a small part of the required computations. Because the analyzed circuits are very simple, the influence of the representation of symbolic functions on the performance of the extractor is difficult to observe. However, for more general applications of symbolic simulation (more complex



(a)



(b)

Figure 11.14 Extracted transistor parameters in function of bias voltages: (a) dynamic G_m; (b) $R_{ds} = 1/G_{ds}$.

circuits and/or greater numbers of symbols), the symbolic functions become very complex, and then their representation must be carefully selected. The simplification of symbolic functions also should be considered [42], [43].

The implementation of symbolic analysis in the FIT-S program was done with emphasis on efficient ac analysis of linear, lumped, and stationary circuits. For such circuits, the characteristic functions are polynomials in the complex frequency (s) with real, constant coefficients. Some modifications are needed if distributed elements (e.g., transmission lines modeling accesses to measured devices) or other frequency-dependent elements are to be taken into account.

In the present implementation, symbolic analysis is applied to ac analysis only. However, results of ongoing research on extensions of the symbolic approach to other types of analyses (nonlinear, transient) ([42], [44], and chapter 7 of this book) could be of interest. The modular structure of the FIT-S program should be helpful in implementation of such enhancements.

References

[1] W. M. Zuberek et al., "Simulation-based parameter extraction and some applications," *IEE Proc.Circuits, Devices Syst.*, vol.141, no. 2, pp.129–134, April 1994.

[2] A. Ibarra and J. Gracia, "Strategy for DC parameter extraction in bipolar transistors," *IEE Proc. Circuits, Devices Syst.*, vol. 137, no. 1, pp. 5–11, January 1990.

[3] F. Lin and G. Kompa, "FET model parameter extraction based on optimization with multiplane data-fitting and bidirectional search—a new concept," *IEEE Trans. Microwave Theory Tech.*, vol. 42, no. 7, pp. 1114–1121, 1994.

[4] C. Scharff, J. C. Carter, and A. G. R. Evans, "New and fast MOSFET parameter extraction method," *Electron. Letters*, vol. 28, no. 21, pp. 2006–2008, 1992.

[5] P. R. Karlsson and K.O. Jeppson, "An efficient parameter extraction algorithm for MOS transistor models," *IEEE Trans. Electron Devices*, vol. 39, no. 9, pp. 2070–2076, 1992.

[6] K. Doganis and D. L. Scharfetter, "General optimization and extraction of IC device model parameters," *IEEE Trans. Electron Devices*, vol. 30, no. 9, pp. 1219–1228, 1983.

[7] P. Conway et al., "Extraction of MOSFET parameters using the simplex direct search optimization method," *IEEE Trans. Computer-Aided Desig*n, vol. 4, no. 4, pp. 694–698, 1985.

[8] K. Garwacki, "Extraction of BJT model parameters using optimization method," *IEEE Trans. Computer-Aided Design*, vol. 7, no. 8, pp. 850–854, 1988.

[9] J. W. Bandler et al., "Integrated model parameter extraction using large-scale optimization concepts," *IEEE Trans. Microwave Theory Tech.*, vol. 36, no. 12, pp. 1629–1638, 1988.

[10] A. Davies and A. K. Jastrzebski, "Parameter extraction technique for nonlinear MESFET models," *Proc. IEEE Conf. Microwave Theory Tech.*, pp. 747-750, 1990.

[11] L. T. Wurtz, "GaAsFET and HEMT small-signal parameter exraction from measured S-parameters," *IEEE Trans. Instrumen. Meas.*, vol. 43, no. 4, pp. 655–658, 1994.

[12] S. Lee and S. W. Kang, "A parameter extraction method using cutoff measurement for a large-signal HSPICE model of HBTs," *IEEE Trans. Electron Devices*, vol. 41, no. 1, pp. 112–114, 1994.

[13] M. Eron et al., "MESFET model extraction and verification techniques for nonlinear CAD applications," *Proc. Third Asia-Pacific Microwave Conf.*, Tokyo, Japan, pp. 321–324, 1990.

[14] G. L. Bilbro et al., "Extraction of the parameters of equivalent circuits of microwave transistors using tree annealing," *IEEE Trans. Microwave Theory Tech.*, vol. 38, no. 11, pp. 1711–1718, 1990.

[15] Y. H. Hu and S-W. Pan, "SaPOSM: An optimization method applied to parameter extraction of MOSFET models," *IEEE Trans. Computer-Aided Design*, vol. 12. no. 10, pp. 1481–1487, 1993.

[16] S. G. Skaggs et al., "Parameter extraction of microwave transistors using a hybrid gradient descent and tree annealing approach," *IEEE Trans. Microwave Theory Tech.*, vol. 41, no. 4, pp. 726–729, 1993.

[17] A. Konczykowska, W.M. Zuberek, and J. Dangla, "Parameter extraction with the FIT-2 program," *Proc. 10th European Conf. Circuit Theory and Design*, vol. 2, Copenhagen, Denmark, pp. 762–771, 1991.

[18] *HP IC-CAP Modeling and Characterization Software*. Hewlett-Packard Information, Santa Rosa, CA, 1993.

[19] *HarPE, User's Manual*. Optimization System Associates, Dundas, Ontario, 1990.

[20] *SUXES 20, User's Manual*. Electrical Engineering Software, Inc., Santa Clara, CA, 1988.

[21] *UTMOST*. SILVACO Information, 1992.

[22] W. M. Zuberek, "SPICE-PAC, a package of subroutines for interactive simulation and circuit optimization," *Proc. IEEE Int. Conf. Computer Design*, Port Chester, NY, pp. 492–496, 1984.

[23] E. Cohen, *Program reference for SPICE 2*. Memorandum UCB/ERL M592, University of California, Berkeley, 1976.

[24] H. Wang et al., "Temperature dependence of DC currents in HBT," *Proc. IEEE Int. Microwave Symp.*, Albuquerque, New Mexico, pp. 731–734, 1992.

[25] A. Konczykowska, W. M. Zuberek, and J. Dangla, "Characterization of semiconductor devices using technological and geometric parameters," *Proc. 35th Midwest Symp. Circuits Syst.*, Washington, DC, pp. 412–415, 1992.

[26] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.

[27] *NAG FORTRAN Library Manual Mark 9*, vol. 3. Numerical Algorithms Group, Oxford, UK, 1982.

[28] W. M. Zuberek and A. Konczykowska, *FIT-S, A Simulation-based Data-driven Parameter Extraction Program*. Tech. Rep. #9402, Department of Computer Science, Memorial University of Newfoundland, St. John's, Canada A1C-5S7, 1994.

[29] D. O. Pederson, "A historical review of circuit simulation," *IEEE Trans. Circuits Syst.*, vol. 31, no. 1, pp. 103–111, 1984.

[30] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New-York: Van Nostrand Reinhold, 1983.

[31] W. J. McCalla, *Fundamentals of Computer-Aided Circuit Simulation*. Boston: Kluwer Academic, 1988.

[32] W. K. Chen, *Applied Graph Theory - Graph and Electrical Networks*. Amsterdam: North-Holland, 1976.

[33] W. Zuberek, A. Konczykowska, and D. Martin, "An approach to integrated numerical and symbolic circuit analysis," *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 1, pp. 33–36, London, UK, 1994.

[34] A. Konczykowska et al., "Symbolic analysis for CAD of microwave circuits," *Symp. Computer-Aided Design Microwave Circuits*, London, 1985.

[35] C. W. Ho, A. E. Ruehli, and P. A. Brennan, “The modified nodal approach to network analysis,” *IEEE Trans. Circuit Syst.*, vol. 22, no. 6, pp. 504–509, 1975.

[36] J. Starzyk and A. Konczykowska, “Flowgraph analysis of large electronic networks,” *IEEE Trans. Circuits Syst.*, vol. 33, no. 3, pp. 302–315, 1986.

[37] C. L. Coates, “Flow graph solutions of linear algebraic equations,” *IRE Trans. Circuit Theory*, vol. 6, pp. 170–187, 1959.

[38] N. Hassaine, “Modelling of passive elements of GaAs integrated circuits.” Ph.D. dissertation (in French), INPG, Grenoble, 1994.

[39] N. Hassaine et al., “Library of parasitic element models for MMICs and MHMICs,” *Proc. 22nd European Microwave Conf.*, Espoo, Finland, pp. 356–361, 1992.

[40] A. Konczykowska et al., “Parameter extraction of semiconductor devices electrical models using symbolic approach,” *Alta Freq.*, vol. 5, no. 6, pp. 291–293, 1993.

[41] A. Konczykowska and W. Zuberek, “Function evaluation in symbolic analysis,” *Proc. IEEE Int. Symp. Circuits Systems*, vol. 3, pp. 2217–2220, Seattle, WA, 1995.

[42] G. Gielen and W. Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*. Boston: Kluwer Academic, 1991.

[43] F. V. Fernandez et al., “Formula approximation for flat and hierarchical symbolic analysis,” *Int. J. Analog Integrated Circuits and Signal Processing*, vol. 1, pp. 183–208, November 1991.

[44] B. Alspaugh and M. Hassoun, “A mixed symbolic and numeric method for closed-form transient analysis,” *Proc. 10th European Conf. Circuit Theory and Design*, Davos, Switzerland, pp. 1687–1692, 1993.