applied sciences

MDPI

Article

A SPICE-Oriented Method for Finding Multiple DC Solutions in Nonlinear Circuits

Stanisław Hałgas

Department of Electrical, Electronic, Computer and Control Engineering, Lodz University of Technology, Stefanowskiego 18/22, 90-924 Łódź, Poland; stanislaw.halgas@p.lodz.pl

Featured Application: Identifying the presence/absence of multiple operating points is especially important in critical systems used in automotive, healthcare, and military applications to eliminate the undesired points and ensure proper circuit operation. The proposed method allows the determination of multiple operating points for a broad class of electronic circuits. Thus, it can be useful in future EDA software for electronic circuits.

Abstract: Finding multiple operating points of nonlinear circuits is a fundamental issue in electronic circuit analysis and design. Methods that guarantee finding all DC solutions are very time-consuming and can be used to analyze only rather small-sized circuits. On the other hand, there exist approaches that can find multiple operating points, but they do not guarantee finding all solutions. The methods are less time-consuming, do not require much computing power, and allow for analyzing more complex circuits. This paper proposes an efficient method for determining multiple and, in most cases, all DC solutions. The approach uses a deflation technique, and the SPICE simulator is applied to the calculations. For this purpose, the original nonlinear circuit during the computational process was modified appropriately after successive solutions were determined. The method was verified on benchmark nonlinear circuits from the literature, made in various technologies and using various models. The results of the DC analysis of several circuits are provided.

Keywords: DC analysis; deflation technique; multiple operating points; nonlinear circuits; SPICE simulator

check for updates

Citation: Hałgas, S. A SPICE-Oriented Method for Finding Multiple DC Solutions in Nonlinear Circuits. *Appl. Sci.* **2023**, *13*, 2369. https://doi.org/10.3390/app13042369

Academic Editor: Alessandro Lo Schiavo

Received: 15 January 2023
Revised: 7 February 2023
Accepted: 10 February 2023
Published: 12 February 2023

Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Since the 1970s, with the development of integrated circuit technology and semiconductor components, there has been a growing interest in designing nonlinear circuits and systems and the phenomena accompanying them [1,2]. The paper [1] defined the term nonlinear technology and indicated that it includes nonlinear devices, as well as numerical methods and strategies. Nonlinear technology, especially developed in the 21st Century, aims to design new devices and equipment that can achieve added values over those designed according to linear strategies [1]. In [1], several aspects of nonlinear technology were discussed regarding selected papers published in the journal *Energies* over the past two years. Chaos theory deals with nonlinear systems that exhibit complex, seemingly random behavior. The theory is an interdisciplinary area of research and has significance in many fields of science. High sensitivity to initial conditions, emergent behavior, and bifurcations are peculiar features found in nonlinear systems [2]. The paper [2] pointed out the steadily increasing trend of papers related to chaos over the past few years and discussed 21 papers in this field from the past two years that the authors of the Editorial considered particularly relevant to the field.

The first task in simulating the behavior of a nonlinear electronic circuit is to find the DC operating point. The simulation is important because the operating point is required for other circuit analyses. For example, the DC operating point is used as a starting point for

transient analysis and to determine the parameters of small-signal models in AC analysis. When performing DC analysis of dynamic nonlinear circuits, all capacitors and inductors are removed from the original system. The inductor terminals are shorted, and the capacitor terminals are left open [3,4].

Electronic analog circuits contain nonlinear elements such as diodes and transistors. These semiconductor devices are described by many nonlinear equations and are characterized by many parameters. The newer the technology, the more complex the description of a single device. Therefore, the general circuit description leads to a large number of nonlinear equations. The robust and efficient calculation of the solutions to a set of nonlinear equations is a theoretically and practically difficult problem. The calculation of the DC operating points of a nonlinear circuit and the so-called power flow solutions of a power grid are among the most-important tasks in electronic and electrical engineering. Certain circuits, such as amplifiers and logic gates, are designed to have a unique DC operating point. However, circuits such as flip-flops, static shift registers, static random access memory (RAM) cells, latch circuits, line receivers, Hopfield neurons, negative resistance circuits, and Schmitt triggers possess multiple isolated DC operating points [5,6]. There are also many important circuits containing the above circuits as sub-circuits [6]. Furthermore, transconductance-amplifier-based filters (G_m-C filters) and most log-domain filters use negative differential resistances produced by positive feedback to increase the quality factor of the integrator, which can lead to the appearance of an undesirable operating point (or points) and make the circuit unusable [7]. Even simple circuits fabricated in MOS/CMOS technology can have multiple solutions. The paper [8] presented a two-transistor MOS circuit with five operating points. An open-loop analysis showed that simple tristable latches could have five operating points [9]. Multiple operating points sometimes result in circuits such that the systems do not operate as designed. In such a case, verifying the presence of multiple DC solutions before the production of the circuits is essential [10]. Determining the solution at equilibrium is also an important issue when studying chaos.

Finding multiple DC solutions of nonlinear resistive circuits is a difficult task. Although it is known that some circuits with positive feedback can maintain multiple operating points, it is not possible to reliably predict multiple DC solutions just by inspecting the circuit topology and component values. Since the standard DC analysis of a circuit in the SPICE software does not automatically detect the possibility of multiple operating points, it is essential to look for them systematically. As shown in the paper [7], the number of operating points in a simulation depends on how the circuit is modeled and may require detailed modeling in regions of operation that are often not accurately modeled. The presence of positive feedback with a loop gain greater than unity is a necessary (though not sufficient) condition for equilibrium point instability [7].

The Newton method and its variants are widely used to calculate solutions to nonlinear equations, for example, in circuit and power grid simulators. These methods can achieve quadratic or super-linear convergence, provided that an initial guess close enough to the solution is specified. However, the methods have some fundamental disadvantages. First, convergence is only guaranteed if the user can provide a sufficiently accurate initial guess. In addition, the lack of convergence of the method does not imply the lack of circuit solutions. Finally, once a solution is found, the methods, due to the specific nature of the numerical algorithm used, do not provide information on the existence of other solutions. Thus, the methods are unsuitable for computing multiple solutions [11]. Another problem is that we have to recalculate the Jacobian matrix at the beginning of each iteration. This matrix contains all the partial derivatives of the nonlinear equations with respect to the circuit variables, which is computationally expensive [3]. Algorithms based on the Newton method, with the default settings of the simulator, often find a solution corresponding to the unstable equilibrium point, e.g., for the circuit shown in Figure 1 using IsSpice4 [12,13], such a solution was determined (the nodal voltages are shown in Figure 1).

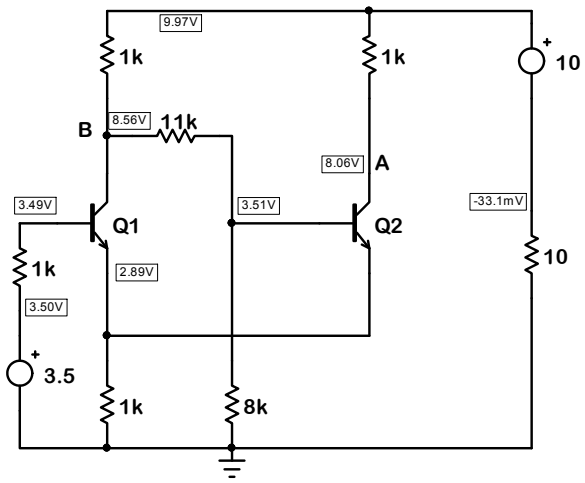


Figure 1. Schmitt trigger and the results of default DC analysis in IsSpice4 software.

The SPICE circuit simulator has become an industry standard, and many SPICE-like tools are now in use. Various efficient techniques such as sparse matrix techniques, LU decomposition, modified Newton's method, and stable numerical integration methods are implemented in SPICE. In addition, SPICE contains various know-how accumulated over 50 years. Therefore, in the field of circuit simulation, from a practical point of view, it seems a good practice to develop a new method using SPICE as the core [14]. Therefore, this paper also decided to follow this approach. Typically, SPICE-simulator-based programs offer sets of different methods rather than a single method for DC analysis, allowing one to choose a new procedure when a tried algorithm fails. For example, MicroCap 10 offers five basic methods: standard Newton method, source stepping, diagonal Gmin stepping, junction Gmin stepping, and pseudo transient. The user controls the methods and the order in which they are tried. IsSpice4 offers three methods: standard Newton, source stepping, and Gmin stepping [15]. Since traditional methods of solving nonlinear equations describing transistor circuits have often shown convergence difficulties, more sophisticated mathematical techniques and tools, such as parameter embedding, continuation, and homotopy methods, have been implemented. Some of these techniques are discussed later. These methods make it possible to analyze circuits that could not be simulated using conventional techniques and can also be used to find multiple DC operating points [5].

Finding all operating points (DC solutions) of a nonlinear circuit is a fundamental problem in the computer-aided design of analog circuits [4,16–21]. Many approaches have been proposed to solve the problem. The most-commonly used methods to find all DC solutions are those based on piecewise linear (PWL) approximation and computational techniques [4,14,17,18,22]. PWL approximation of the nonlinear characteristics of circuit elements allows the problem to be solved by checking some regions (or hyper-regions) for solutions. However, as shown in numerical experiments [18], the PWL approximation often changes the number of solutions. Thus, to design a high-reliability circuit, it is necessary to find all solutions to the original nonlinear circuits. For this purpose, interval methods have been developed. The methods are capable of finding all DC solutions at a given accuracy [16,19,21,23,24]. However, the computational time of interval algorithms increases exponentially with the number of variables. Introducing various tests to interval algorithms (e.g., linear programming test) allows all DC solutions to be determined more efficiently [18,19]. A similar approach is based on the idea of successive contraction, division, and elimination of certain hyper-rectangular regions in which solutions are sought. The essential point of this approach is the contraction method, which leads to obtaining bounds on the location of all solutions inside a hyper-rectangular region [20,22]. Although all the methods are mathematically elegant, they are rather time-consuming and computationally intensive for large-scale circuits. Consequently, only relatively small or

specific nonlinear circuits can be effectively analyzed. In addition, these methods usually require a specific hybrid description, apply to circuits belonging to a strictly defined class, and use simple models of semiconductor elements (e.g., the Ebers–Moll model of a bipolar transistor, the Ebers–Moll-type model (The Ebers–Moll model of bipolar transistors is composed of two diodes and two current-controlled current sources. The MOS transistors with a long channel can be characterized by a model having the same structure, called the Ebers–Moll-type model. This model is equivalent to the Shichman–Hodges model with the omission of some secondary phenomena [25,26]) or the Shichman–Hodges model of an MOS transistor) [19,22,27].

The theoretical problem of finding all solutions to an electronic circuit of significant size is still unsolved and, according to many researchers, will remain unsolved and challenging for a long time to come [28]. Therefore, some methods focus on determining only a certain subset of these solutions. Many homotopy methods have been developed to calculate multiple solutions [5,6,11,14,27,29–32]. Most homotopy methods follow the multi-start or extended curve tracking (lambda-threading) approach. The multi-start approach selects multiple initial guesses at the homotopy parameter, lambda, equal to zero and follows the continuation paths to solutions at lambda = 1. The extended curve-tracing approach follows a single homotopy path past lambda = 1, hoping to reverse direction and pass through lambda = 1 multiple times. This approach may not find all solutions if the tracked path does not pass through all solutions, but it is efficient because it can avoid revisiting the same solution [29]. For example, Reference [6] developed a SPICE-oriented Newton homotopy method based on the arc length approach, which chooses one of the node voltages in each positive feedback loop as a variable and traces the solution path using SPICE transient analysis. The work [27] brought a method using thread homotopy based on transistor states. The papers [11,29] provide a comprehensive analysis of homotopy methods for calculating all DC solutions of nonlinear circuits and systems. Several sufficient conditions were derived to guarantee finding all solutions along a single homotopy path. An efficient method for finding operating points of nonlinear circuits containing neither voltage- nor current-controlled PWL resistors using the homotopy method and SPICE-oriented approach was proposed in [14]. Finding multiple DC operating points in memristor circuits using homotopy methods was studied in [31]. The paper [32] brought a numerical method for determining multiple DC solutions based on the concept of separable Newton homotopy and partitioning the solution space into currently changing rectangles. In the literature, also papers applying a different approach to determining multiple DC solutions can be found. For example, the paper [3] discussed an approach based on evolutionary computation for determining multiple solutions using parallel search. The paper [33] presented a procedure for approximating DC operating points of nonlinear circuits based on the Carleman linearization. The idea of deflation combined with the discrete circuit equivalent of the Newton method was used to search for multiple solutions in bipolar [34] and short-channel CMOS circuits [15]. The methods usually allow multiple DC solutions to be determined, but cannot guarantee that all operational points will be found. Their implementation is generally tedious, and their use often involves specific classes of circuits.

Identifying circuits with multiple DC solutions can be solved in two ways: finding more than one operating point by applying the methods described above or by searching for some topological structure. A good design tool should indicate the possible presence of more than one operating point [28]. A circuit can be claimed to have multiple solutions when more than one operating point is found [10,28]. The approach presented in [28] is based on dividing the original circuit into sub-circuits and building an oriented dependency graph. If the graph has oriented loops, they can be sources of multiple operating points. The loops are opened by removing a minimum number of circuit nodes. Next, multiple solutions are searched, sweeping specific voltage sources over appropriate intervals. A systematic method for automatically identifying positive feedback loops in analog/mixed-signal CMOS circuits was proposed in [10,35,36]. The method first transforms a circuit netlist into a directed dependence graph that captures the critical relationships between

branch currents and node voltages. Next, all feedback loops are found, and finally, some criteria are applied to determine the positive feedback loop. Moreover, paper [36] proposed a divide and contraction verification method against undesired operating points in CMOS circuits. Unlike traditional methods for finding all operating points, this method only targets the search for voltage ranges containing operating points.

The main objective of this work was to develop a method that allows the finding of multiple solutions in a broad class of nonlinear circuits. In addition, the method should be fast, and its use should not require preliminary steps by the designer, e.g., identifying positive feedback loops, selecting an initial guess, or creating a hybrid description. The successful operation of the method should be independent of the engineer's knowledge, and the semiconductor devices should be described with adequate nonlinear equations corresponding to the technologies used. Therefore, applying the method using the SPICE software as the computational core was decided. In order to circumvent the limitations of the software resulting from the Newton algorithm used in the DC analysis (with built-in improvements), the circuit was modified by introducing nonlinear dependent current sources implementing the deflation idea discussed in Section 2. The method can be applied using a graphical user interface (ICAP/4 environment [12] in the proposed approach). However, a more efficient approach was to develop a host application that generates the appropriate netlists for IsSpice4 [13] and processes the simulation results. Both approaches are described in Section 3. In contrast to existing SPICE-oriented methods, the proposed method does not use time domain analysis, but typical DC analysis. It can be used to analyze circuits at the level of bipolar and MOS transistors (with models of long- and short-channel MOSFETs) and circuits with macromodels, e.g., of operational amplifiers. In order to illustrate the effectiveness of the method, a database of 35 nonlinear circuits with multiple DC solutions was developed. Although the method does not guarantee the finding of all DC solutions, in many cases, it finds them. Therefore, the proposed approach is a valuable tool that not only gives the designer an indication of the possibility of undesired solutions, but also determines them quickly. In order to confirm which of the numerically found solutions occur in real circuits, two circuits were built, and their solutions were measured in the laboratory. The results of the numerical simulations and the laboratory tests are discussed in Section 4. The research is summarized in Section 5.

2. Deflation Technique

The approach proposed in the paper uses a concept known in mathematics as deflation [37–41]. The theoretical foundations of deflation methods were discussed in [37]. In [38], theoretical considerations were extended to the case of infinite-dimensional Banach spaces, allowing deflation techniques to be applied to systems of partial differential equations. Moreover, new classes of deflation operators were introduced. The purpose of paper [39] was to present a deflation algorithm for finding the roots of a system of nonlinear equations where the Jacobian matrix is singular at a root. A systematic and automatic way to reduce the degree of the multiplicity of roots was proposed. A deflation method to find the successive eigenvalues in the numerical simulation of the band structure of three-dimensional dispersive metallic photonic crystals with face-centered cubic lattice was presented in [40]. In [41], the continuation Newton method with the deflation technique and the quasi-genetic evolution for the global minimum searching of an optimization problem was established.

One of the interesting problems when solving a nonlinear equation is the existence of a magnetic zero [37]. The magnetic zero is the zero (root) to which the algorithm converges almost independently of the initial guess selected. Such a magnetic zero can often mask zeros of real interest. A particular advantage of deflation techniques is the possibility of avoiding convergence to such a magnetic zero once it has been determined.

The basis of the deflation method is the following technique [37]. After determining the root, a new equation (system of equations) is created so that it retains those roots of the original equation (system) that remain to be calculated, but the previously found root

is no longer a solution of the equation. The deflation technique was studied initially in the one-dimensional case. Let $f(x)$ be a scalar polynomial and $x^1, \cdots, x^k$ be roots of $f(x)$ identified with some iterative algorithm, such as the Newton method using, e.g., different initial guesses. Different roots of $f(x)$ may be found by considering the deflated function:

$$g(x) = \frac{f(x)}{(x - x^1) \cdot (x - x^2) \cdot \ldots \cdot (x - x^k)} \tag{1}$$

and applying the same iterative algorithm to $g(x)$ as to $f(x)$.

The paper [37] introduced the concept of the deflation matrix, developed deflation theory for systems of nonlinear equations, and extended it to the problem of determining multiple roots. Several specific deflation techniques were proposed, and their application to function minimization was shown. Let us consider a nonlinear equation:

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}, \tag{2}$$

where $\mathbf{x} = [x_1, \ldots, x_n]^{\mathrm{T}}$, $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})]^{\mathrm{T}}$, and T denotes the transpose. Suppose that (2) has several solutions and one of them, labeled $\mathbf{x}^1$, has already been found. To determine other solutions, Equation (2) is deformed to retain the remaining solutions, but avoids $\mathbf{x}^1$. In [37], the following deformation was proposed. Let $\mathbf{M}\left(\mathbf{x}, \mathbf{x}^1\right)$ be an n by n matrix, called a deflation matrix, such that

$$\liminf_{i \to \infty} ||\mathbf{M}\left(\mathbf{x}^{(i)}, \mathbf{x}^1\right)\mathbf{f}\left(\mathbf{x}^{(i)}\right)|| > 0 \tag{3}$$

for any sequence $\mathbf{x}^{(i)} \to \mathbf{x}^1$. Let us create the function:

$$\mathbf{g}(\mathbf{x}) = \mathbf{M}\left(\mathbf{x}, \mathbf{x}^1\right)\mathbf{f}(\mathbf{x}). \tag{4}$$

We solve Equation (4) using a method of determining a single solution (e.g., the Newton method). Such a choice of matrices that satisfy Condition (3) ensures that $\mathbf{x}^1$ is not a solution of Equation (4) and that the sequence generated by the method will converge to another solution of the original Equation (2) or diverge. In order to deflate k solutions $\mathbf{x}^1, \ldots, \mathbf{x}^k$, we formulate the deflation function in the form:

$$\mathbf{g}(\mathbf{x}) = \mathbf{M}\left(\mathbf{x}, \mathbf{x}^1\right) \cdots \mathbf{M}\left(\mathbf{x}, \mathbf{x}^k\right)\mathbf{f}(\mathbf{x}). \tag{5}$$

Two methods of deflation, norm deflation and inner product deflation, differing in the form of a matrix $\mathbf{M}$, were proposed in [37]. The deflation matrix used in norm deflation has the form:

$$\mathbf{M}\left(\mathbf{x}, \mathbf{x}^j\right) = \frac{1}{||\mathbf{x} - \mathbf{x}^j||}\mathbf{A}, \tag{6}$$

where $\mathbf{A}$ is an $n \times n$ matrix. In the inner product deflation, the matrix $\mathbf{M}$ is assumed to be in the form of a diagonal matrix, whose i-th diagonal element ($i = 1, \ldots n$) is given by

$$m_{ii} = \frac{1}{\langle \mathbf{a}_i^j, \mathbf{x} - \mathbf{x}^j \rangle}, \tag{7}$$

where $\mathbf{a}_i^j$ is an n-dimensional vector.

Unfortunately, the described approaches are ineffective in the DC analysis of electronic circuits [15]. In the paper [34], deflation was used to analyze diode–transistor circuits in which the transistors were characterized by the Ebers–Moll model. The proposed deflation technique, combined with the discrete circuit equivalent to the Newton method, makes it possible to find multiple DC solutions. An extension of this concept to the analysis of CMOS transistor circuits using the PSP 103.1.1 model of the MOS transistor was proposed in [15]. The deflation technique was combined with the homotopy approach.

Although both methods are effective, their application is limited to specific classes of circuits. Moreover, they require writing custom software and preselecting specific parameters in the computational process. Therefore, the following section proposes another way to apply the concept of deflation to the problem of finding multiple DC solutions. An approach was developed using the standard SPICE software as the core. The method makes it possible to find solutions in circuits made of discrete elements and integrated circuits manufactured in various technologies.

3. SPICE-Oriented Deflation Method

ICAP/4 is the circuit simulation software package from Intusoft [12,13]. SpiceNet is a schematic input program that allows placing circuit elements by drag and drop and setting their attributes. The schematic is saved in a file with the .dwg extension. The schematics created in SpiceNet for the four circuits are shown in Figures 1 and 3–8. SpiceNet allows an analysis wizard to select the desired analysis, the required parameters, and the simulator options. It generates SPICE netlists, which are necessary to run a circuit simulation. The netlist is a standard text file containing information regarding the description of elements, models used, selected analyses, and options. All netlists must have a title line and a .END line. IsSpice4 loads the netlist and runs the simulations. DC analysis determines a circuit's quiescent DC operating point with shorted inductors and open capacitors. The output file from IsSpice4 is compatible with output files generated by Berkeley SPICE Version 2. Data are stored in the same tabular and printer-plot formats [13].

IsSpice4 is based on the Berkeley SPICE 3F.5 kernel. The analog behavioral model (ABM) capabilities in IsSpice4 increase the flexibility in describing electronic, mechanical, and physical systems. The ABM functions are implemented using linear dependent sources (key letters E, F, G, or H) and nonlinear dependent sources (key letter B). The nonlinear dependent source allows the use of algebraic and trigonometric operators (including, i.e., exp(x)—exponential function, min(x,y)—minimum of x and y, and sqrt(x)—square root), node voltages, and currents in the description. It is also possible to use the conditional function if–then–else (format: condition? output (if the condition is true): output (else)).

The convergence properties of DC analysis in IsSpice4 have been improved as compared to SPICE 3F.5. In addition to automatically calling the traditional source stepping algorithm, IsSpice4 includes an additional algorithm called Gmin stepping. This algorithm uses a minimum junction conductance, which keeps the sparse matrix well conditioned, and a separate variable conductance to ground at each node, which serves as a DC convergence aid. IsSpice4 selects this procedure automatically when convergence problems arise. Source stepping is called when both the default method and the Gmin stepping algorithm fail. Source stepping sets all sources to a value close to zero to make it easier to compute a DC solution. When a solution is found, the source values are increased toward their nominal values. Each subsequent operating point is calculated using the previous solution as the starting point. This process continues until the sources reach their nominal values [13]. Suppose the node voltages do not settle within a certain number of iterations. In that case, the DC analysis will issue an error message such as No convergence in DC analysis, Singular Matrix, or Gmin/Source Stepping Failed (see Figure 5).

There are two options in IsSpice4, with values and flags. The options are specified in the netlist. Two of them were used in the implementation of the deflation method. Option ITL1=A sets the limit A to the number of Newton iterations that IsSpice4 will perform before declaring No convergence in DC analysis (value A = 1000 is used in the proposed method). When reaching this limit without convergence, IsSpice4 will automatically invoke the built-in Gmin stepping and source stepping algorithms to achieve convergence. Entering option RSHUNT=B results in the value (B) used as a shunt resistance to ground from each node [13] (value B = 10^{12} Ω is used in the method).

After this brief introduction to the capabilities and options of IsSpice4, the proposed method for finding multiple DC solutions will be discussed in detail. Two versions are presented. The first version uses the graphical environment offered by the SpiceNet

program from the ICAP/4 software suite. However, user intervention is required to enter the parameters of behavioral models, which depend on the previously determined solutions. The second version is a fully automated procedure operating on netlists, created by a host application written in the Delphi environment. All calculations are performed in IsSpice4, while the host application is responsible for creating batch files and processing the results.

Once the first DC solution of the considered circuit has been found using the standard DC analysis procedure, an essential element of both versions is to modify the schematic to eliminate the possibility of recalculating this solution by using the concept of deflation. The assumption is that the auxiliary circuit connected to the original circuit should be simple to implement and that the built-in DC analysis will still be used. The limitation is that only the elements and functions available in SPICE can be utilized. As a result of preliminary research, it was decided to use nonlinear dependent current sources (B sources). The number of these sources is two in the case of searching the second solution and $k+1$ for the third and subsequent solutions, where k is the index of solutions sought. The fundamental element is the source labeled BC (see Figures 3–5). The labeling of the dependent source with the abbreviation BC is due to the fact that the first letter is the designation required by the IsSpice software for a nonlinear controlled source, and the second letter denotes that it is the current source attached to the selected circuit node. The source BC is present in every schematic (netlist) and is, thus, not indexed like the other B-type sources. It is attached to a selected node of the circuit. When the circuit is supplied from a real voltage source or sources (i.e., a source with non-zero internal resistance), the external terminal of the source is selected as the connection node (see Figures 3–5). When the schematic is built using ideal voltage sources, we select any node except the ungrounded nodes of the sources. The value of the BC source current depends on the Euclidean norm of the difference between the solution in a current iteration and the previously determined solution (for $k = 2$) or the minimum Euclidean norm of the difference between the solution in a current iteration and the previously determined solutions (for $k > 2$). When approaching solutions already found, the current of the source should increase to infinity. However, for new solutions, the current of the source should be zero. Since the goal of the work was to create an algorithm that works in the IsSpice4 software, the selection of functions was limited to those available in the software. Preliminary studies were conducted using various exponential, hyperbolic, and trigonometric functions with different parameters. Using the hyperbolic function a/x, for $x > 0$, sometimes led to errors in the simulation due to overflow. The lack of convergence to solutions close to each other caused an additional problem. It was therefore decided to use the function:

$$f(x) = a \cdot \mathrm{e}^{-b \cdot x^4}, \tag{8}$$

where $a = 10^5$ and $b = 10^5$. The plot of the function is shown in Figure 2.

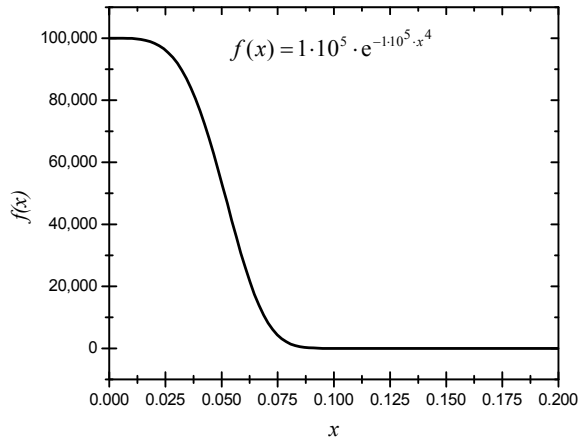


Figure 2. The plot of the function used in the dependent current source (BC source).

Based on numerous simulation studies, the proposed function with the given values of parameters avoided many of the problems present with other functions, such as divergence, slow convergence, and losing solutions. The function in the iterative process enters saturation, at enormous current values, for solutions close to the solutions already determined. The function did not lead, in any case, to breaking the process due to numerical overflow. In addition, at $x = 0.15$, the function takes a value 1×10^{-17}. Above $x = 0.15$, we took 0 to avoid numerical problems (see e.g., Figure 3). Therefore, it is possible to determine solutions not close to each other in the Euclidean norm than 0.15. The modification of the function, for example, taking $a = 10^3$ and $b = 10^3$ solutions not close, then 0.3 can be found. In such a case, in circuits with low supply voltage, some solutions may not be found. Taking greater values of the parameters a and b can lead to difficulties in converging the numerical process. The choice of the values of the parameters could be subjected in the future to an optimization process aimed at maximizing the number of determined solutions for all test examples combined. It is also possible to make them dependent on the supply voltage and the class of circuits under consideration.

The Bi sources ($i = 1, \ldots, k-1$) determine the Euclidean norms of the difference between the current solution at a given iteration and the previously determined operating points. The BMIN source calculates the minimum value of these distances (for $k > 2$). Creating modified circuits continues until an error is obtained in the DC analysis. Such an error means the impossibility of determining the next operating point and indicates that there is no such point and all DC solutions have already been determined or the procedure fails. Since the method does not guarantee finding all solutions, such a case is, of course, possible. We can then repeat the process by selecting another node to connect the BC source. If, on the other hand, we only care about whether the circuit has multiple operating points and two or more solutions are found, then we stop. Suppose we are interested in determining all the DC solutions. In that case, we can apply the time-consuming procedures described in Section 1 if such a procedure exists for the considered class of circuits.

3.1. Finding Multiple DC Operating Points Using Graphical Environment

1. Set $k := 1$. Create the schematic of the circuit to be analyzed in SpiceNet. Introduce from the libraries adequate models of semiconductor devices and/or macromodels of integrated circuits. Adjust the parameters of the models if necessary. Select DC analysis. Set the options mentioned above. Label the nodes of the original circuit with numerical values. Run the simulation. The results of the activities for an exemplary circuit are shown in Figure 1. If the process converges, save the k-th DC solution of the circuit. Go to Step 2. If the process is divergent, go to Step 5.
2. $k := k + 1$. Select the node to which the nonlinear dependent current source BC will be connected (by default, we assumed the ungrounded terminal of the real voltage source). Connect a B-type element to the node. Create the circuit of the second B-type element (B1 source) and 1 Ω linear resistor (see Figure 3). Assign a label x to the common node of these elements. Enter the relation defining the current of element B1 corresponding to the Euclidean norm of the difference between the node voltages of the currently analyzed circuit and the DC solution determined for $k = 1$ (see Figure 3). Enter the relation determining the current of the BC element using the function defined by Equation (8) and the instruction if...then...else..., i.e., I = V(x)<0.15? 1e5*exp(-1e5*V(x)*V(x)*V(x)*V(x)). Perform the standard DC analysis (The results are shown in Figure 3). If the process converges and the value of the BC source current is 0, save the k-th DC solution. Change the label x to x1. Attach the circuit formed of the other source of type B (the source BMIN) and the unit resistor to the schematic, and go to Step 2. If the process is divergent, go to Step 5.
3. $k := k + 1$.

4. Connect to the schematic drawn in Step 2, the circuit formed from the B-type element (source Bl, $l = k - 1$), and the 1 Ω linear resistor (see Figure 4 for $k = 3$ and Figure 5 for $k = 4$). Assign the label xl to the common node of these elements. Enter the relation defining the current of the Bl element corresponding to the Euclidean norm of the difference between the node voltages of the currently analyzed circuit and the DC solution determined for $k - 1$. Enter the relation defining the current of the BMIN element using the min operator (see Figure 4 for $k = 3$ and Figure 5 for $k = 4$). Perform the standard DC analysis. If the process converges and the value of the BC source current is 0, save the k-th DC solution of the circuit, and go to Step 3. If the process is divergent (see Figure 5 for $k = 4$), go to Step 5.
5. Stop.

Note 1: If the DC analysis in IsSpice4 fails in Step 1, it is possible to repeat the process after changing the default DC analysis options, e.g., by increasing the ITL6 parameter, which determines the number of steps in the source stepping algorithm. Such a case occurs in none of the circuits considered in the paper.

Note 2: The discussed GUI-based algorithm does not require any additional software. However, the user has to update the schematic each time and enter or modify specific nonlinear dependencies for nonlinear dependent sources. Therefore, the following subsection describes a different approach that is fully automated. This method uses a host application that creates netlists of the analyzed circuits, runs SPICE analyses, processes the output files, and saves all determined DC solutions to a file.

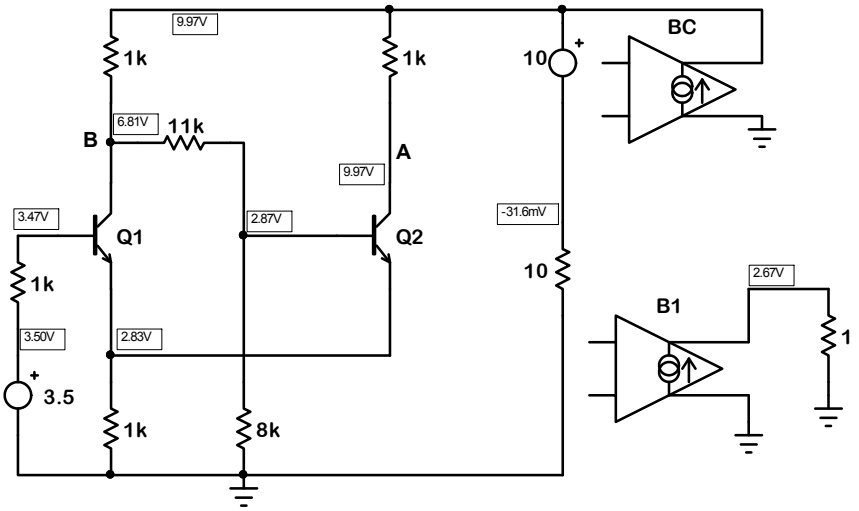


Figure 3. Modified, according to the idea of deflation, circuit shown in Figure 1 and the results corresponding to the second DC solution ($k = 2$).

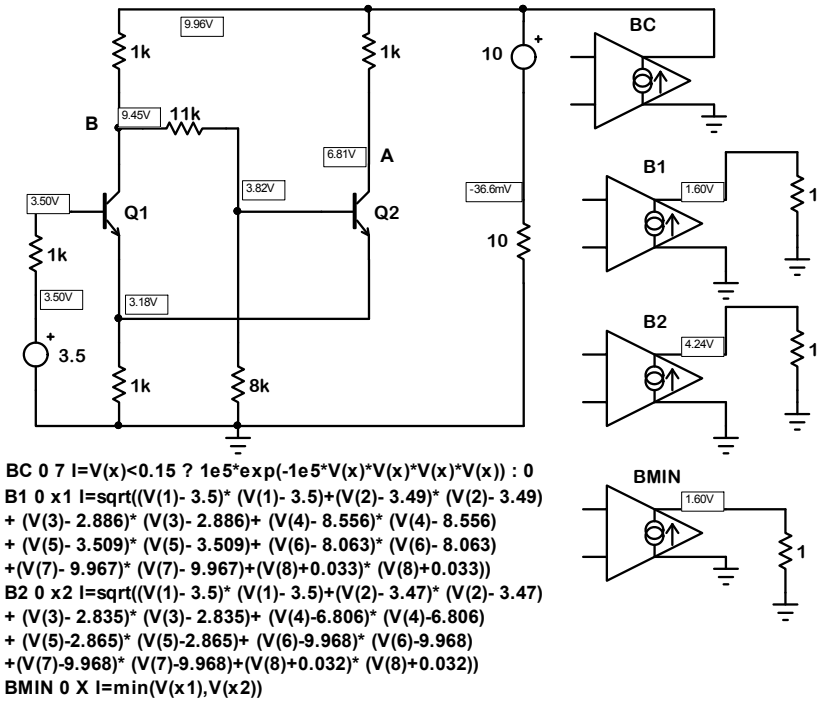


Figure 4. Modified, according to the idea of deflation, circuit shown in Figure 3 and the results corresponding to the third DC solution ($k = 3$).

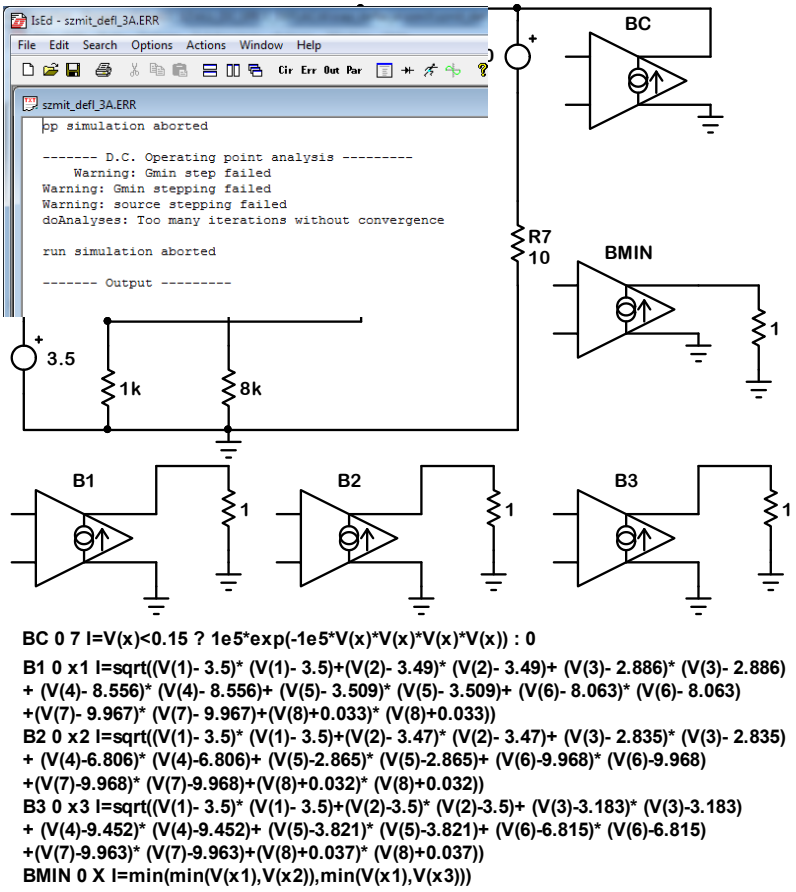


Figure 5. Modified, according to the idea of deflation, circuit shown in Figure 4 with the message of the divergent process ($k = 4$).

3.2. Finding Multiple DC Operating Points Using Batch Processing

The second version of the proposed method is based on the interaction of two programs. The first program is a host application written by the author in the Delphi environment, and the second is the IsSpice4 software (the environment ICAP/4 ver. 8.1.11 Build 3247). In order to link the two environments, the main program uses Delphi's ShellApi library and the ShellExecuteEx function to launch an external application (here, IsSpice4) from the Delphi application in the Windows environment. The user writes a netlist of the circuit to be analyzed as a batch file. Alternatively, a .cir file created in the GUI of the ICAP/4 environment can be used. Next, the host application is launched. It loads the batch file and obtains information about the number of circuit nodes (excluding nodes inside models and macromodels) and the index of the node to which the nonlinear dependent current source (BC) has to be connected. The application then executes Steps 1–5 of the algorithm described in the previous section. It calls the IsSpice4 in a loop, processes the output files to read the necessary data for defining the subsequent dependent current sources, creates the subsequent netlists, saves the solutions, and verifies that the current of the BC source for the designated solution is equal to 0. During the process, the files, after reading the required information, are deleted. Finally, the host application saves the DC solutions and the total calculation time to the output file.

4. Results and Discussion

In order to verify the effectiveness of the proposed method, a database of 35 electronic circuits with multiple operating points was created based on the available literature and the author's own experience. These circuits differ in fabrication technology (bipolar, MOS circuits), transistor models (Ebers–Moll and Gummel–Poon models for bipolar transistors, Level 1, Level 2, and BSIM for MOS transistors), IC macromodels, complexity, and the number of DC solutions. Four of these circuits are shown in Figures 1 and 6–8. The most-important parameters of the circuits and the results are summarized in Table 1. Cases corresponding to the circuits discussed in this work are marked in the *Circuit type* column with the number of the figure where the original schematic of the analyzed circuit is shown.

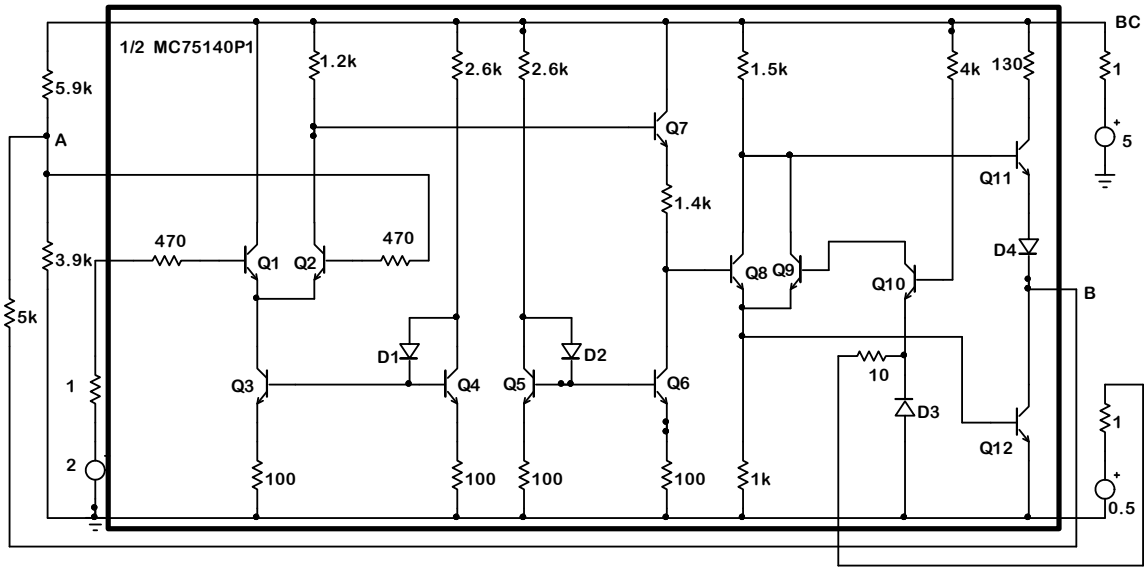
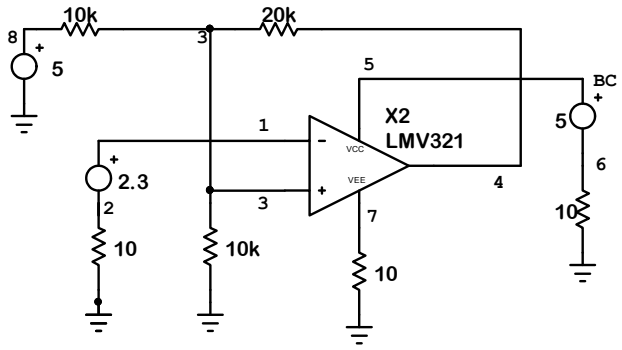
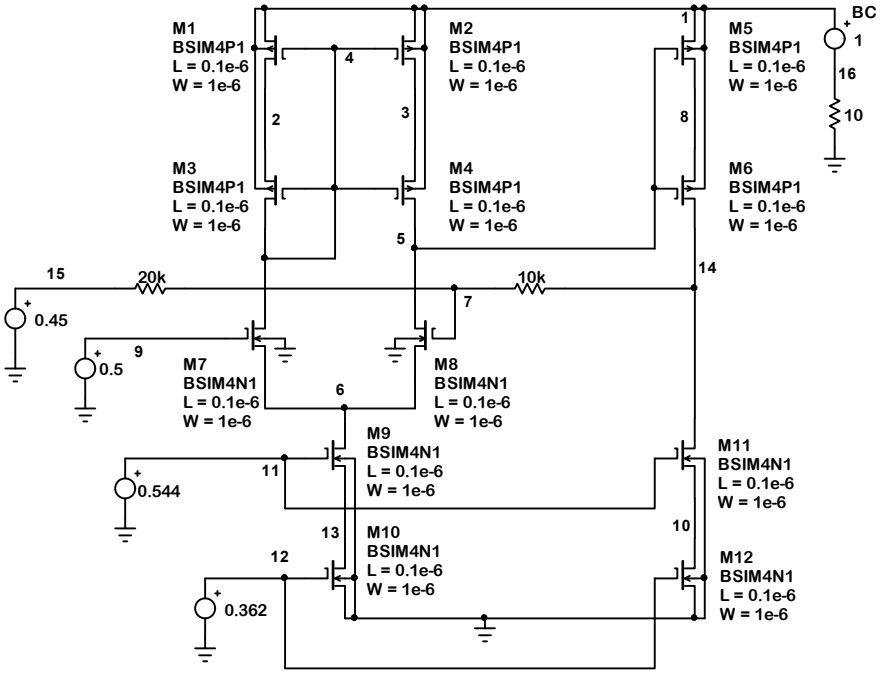


Figure 6. A circuit based on the MC75140 possessing three DC solutions.



Solutions of circuit
Solution 1 2.3 0 2.004 0.021 4.999 −0.001 0.002 5
Solution 2 2.3 0 2.293 1.466 4.999 −0.001 0.002 5
Solution 3 2.3 0 2.994 4.969 4.998 −0.002 0.001 5
Time:1154 ms

Figure 7. A circuit containing an operational amplifier LMV321 modeled using the default IsSpice macromodel and DC solutions found by the proposed method.



Solutions of circuit
Solution 1 1 0.962 0.961 0.600 0.486 0.150 0.505 0.940 0.5 0.151 0.544 0.362 0.116 0.532 0.45 0
Solution 2 1 0.962 0.959 0.602 0.157 0.151 0.716 0.931 0.5 0.159 0.544 0.362 0.116 0.849 0.45 0
Solution 3 1 0.954 0.997 0.554 0.993 0.124 0.370 0.977 0.5 0.146 0.544 0.362 0.100 0.330 0.45 0
Time:1576 ms

Figure 8. CMOS circuit with transistors represented by the BSIM4.6 model and DC solutions found by the proposed method.

Table 1. Key parameters and results of calculations for exemplary circuits with multiple DC solutions.

No.	Number of All DC Solutions	Circuit Type	Number of Macromodels	Number of Transistors *	Number of Diodes *	Number of Found DC Solutions	Time in s
1	9	B [(1)]	0	4	0	9	3.8
2	5	M2 [(2)]	0	2	0	5	1.7
3	3	IC [(3)]	1	0	0	3	1.2
4	3	IC [(4)],Figure 7	1	0	0	3	1.2
5	3	B	0	2	0	3	1.0
6	3	B	0	25	0	3	2.9
7	3	B	0	13	0	3	1.4
8	3	M3 [(5)],Figure 8	0	12	0	3	1.6
9	3	B	0	10	0	3	1.4
10	3	M1 [(6)]	0	14	0	3	1.0
11	3	B	0	13	1	3	2.7
12	3	B	0	56	0	3	4.2
13	3	B Figure 6	0	12	4	3	1.7
14	5	B	0	15	5	5	3.6
15	3	M3	0	12	0	3	1.6
16	5	B	0	5	1	5	2.0
17	3	M1	0	10	0	3	1.3
18	3	M1	0	16	0	3	1.3
19	3	M3	0	16	0	3	1.8
20	3	M1	0	6	0	3	1.1
21	3	M3	0	6	0	3	1.2
22	3	B	0	6	0	3	1.1
23	3	B Figure 1	0	2	0	3	1.0
24	3	M3	0	6	0	3	1.1
25	3	M3	0	5	0	3	1.2
26	3	M3	0	12	0	3	1.3
27	3	M3	0	15	0	3	1.8
28	9	DT [(7)]	0	0	2	5	1.4
29	3	B	0	10	0	3	1.4
30	3	B	0	2	0	3	1.0
31	11	B	0	6	3	5 [(8)]	2.3
32	3	M1	0	8	0	3	1.0
33	5	B	0	7	0	5	1.7
34	3	B	0	9	0	3	1.5
35	3	B	0	4	1	3	1.2

* Without elements used in macromodels; [(1)]—bipolar technology; [(2)]—MOS technology (Model Level 2); [(3)]—circuit with LM318L macromodel (In addition to resistors and linear capacitors, the macromodel contains two bipolar transistors, nine diodes, and six linear controlled sources.); [(4)]—circuit with LMV321 macromodel (In addition to resistors and linear capacitors, the macromodel contains 2 bipolar transistors, 20 diodes and 30 linear controlled sources.); [(5)]—MOS technology (Model BSIM4.6); [(6)]—MOS technology (Model Level 1); [(7)]—a circuit with two tunnel diodes; [(8)]—11 solutions are found after connecting the BC source to another node, in 5.5 s.

The calculations were performed on a computer with an Intel(R) Core(TM) i7-6700 processor and 64GB RAM. The total computation time consists of the time consumed by the algorithm in the Delphi environment, the circuit analysis time performed by IsSpice4, and the time spent on communication between the two environments, including the process of creating new and deleting old files for IsSpice4, searching for output files, and opening and closing windows.

In addition to simulation studies, laboratory tests were conducted, and selected nodal voltages were measured in the two circuits shown in Figures 1 and 6. These tests were carried out by making the circuit using a typical solderless breadboard, the corresponding integrated circuit (MC75140), transistors (BC 107A), and discrete resistors with a 1% tolerance. A 34401A digital multimeter was used to measure the voltages, and the circuits were

powered by a VLP-2403 PRO power supply. As a result of the measurements, voltages corresponding to the two solutions found numerically were obtained for both circuits. At the prototyping stage, the circuit shown in Figure 1 was made of selected BC 107A transistors with the parameter BF being about 100. The BC 107A model was downloaded from the IsSpice4 libraries, and the parameter BF was set to 100. The measured voltage values were $V_A^{(1)} = 9.988\,\mathrm{V}$, $V_B^{(1)} = 6.823\,\mathrm{V}$ and $V_A^{(2)} = 6.781\,\mathrm{V}$, $V_B^{(2)} = 9.480\,\mathrm{V}$, respectively. The values correspond to the solutions shown in Figures 3 and 4, i.e., $V_A = 9.968\,\mathrm{V}$, $V_B = 6.806\,\mathrm{V}$, and $V_A = 6.815\,\mathrm{V}$, $V_B = 9.452\,\mathrm{V}$, respectively. Thus, the solution determined at the default settings of the DC analysis, $V_A = 8.063\,\mathrm{V}$, $V_B = 8.556\,\mathrm{V}$, shown in Figure 1, corresponds to the unstable equilibrium point. The instability is due to the presence of capacitors and inductors, usually nonlinear, present in the semiconductor devices, passive elements, and the parasitic capacitance and inductance of the connections. It follows that the designer, unaware of the problem, obtains a numerical solution that meets the mathematical description, but is unstable. Therefore, it is essential to use procedures that allow the determination of multiple DC solutions, such as the one proposed in this work. For the circuit shown in Figure 6, the measured voltages corresponding to the two solutions are $V_A^{(1)} = 1.345\,\mathrm{V}$, $V_B^{(1)} = 0.061\,\mathrm{V}$ and $V_A^{(2)} = 2.481\,\mathrm{V}$, $V_B^{(2)} = 3.641\,\mathrm{V}$, respectively. Three DC solutions $V_A^{(1)} = 1.345\,\mathrm{V}$, $V_B^{(1)} = 0.100\,\mathrm{V}$, $V_A^{(2)} = 2.493\,\mathrm{V}$, $V_B^{(2)} = 3.597\,\mathrm{V}$, and $V_A^{(3)} = 2.009\,\mathrm{V}$, $V_B^{(3)} = 2.087\,\mathrm{V}$, were determined using the proposed method. The default DC analysis found the solution corresponding to the stable equilibrium point.

Prototyping was performed only to determine which solutions were stable equilibrium points. It was not important to bring the voltages into a perfect agreement since the influence of the accuracy of the available models, component tolerances, temperature, parasitic parameters, and measurement uncertainty causes the voltages in the actual systems to vary from the simulation. However, statistical analyses taking tolerance into account showed that parameter scatter within the assumed limits for both prototyped circuits did not change the number of solutions. Thus, even taking these phenomena into account, the measurements qualitatively reflect the specifics of the circuit, and the (quantitative) measurement results differed slightly from the simulation results.

The solutions, copied from the output file, of the circuits shown in Figures 7 and 8 are placed under the corresponding schematics.

Summarizing the results shown in Table 1, we see that, in all 35 circuits, the existence of multiple solutions was confirmed. For 33 circuits, all DC solutions were determined. For the circuits labeled 28 and 31, only part of the solutions was determined. For the circuit labeled 31, changing the node to which the BC source is attached from the default to another allowed all 11 solutions to be determined. The analysis time, determined by the time of file operations, was short and did not exceed 6.0 s.

5. Conclusions

The paper proposed an original method for finding multiple DC solutions (operating points) in nonlinear electronic circuits. For this purpose, the deflation technique was used, which was implemented in circuit form in the SPICE environment. Unlike other methods that determine multiple solutions, the proposal allows the analysis of circuits made in various technologies using various models and macromodels. In addition, the calculations used a standard DC analysis rather than a transient analysis, as in the majority of methods using the homotopy concept. Although the method does not guarantee finding all DC solutions, it determines them in the most-practical examples. Unfortunately, the inability to view the results of individual iterations implemented in IsSpice4 cannot provide a rule of thumb in which circuits the method may fail. The only thing that can be said is that the existence of more than five solutions (a case rather rare in practice) can lead to a situation where not all solutions are determined. Then, two options are available to use a method that guarantees the determination of all solutions (if any) or repeat the process while attaching the BC source to another node (or other nodes). Some of the

determined DC solutions correspond to the unstable equilibrium points of the associated dynamic circuits. The analysis time is short and acceptable at the design stage. The paper presented two implementation options. The first version uses only the graphical user interface provided by the available version of SPICE, but requires the input of specific elements and their equations by the user. The second version uses batch processing and the custom host application, which controls the computational process automatically. Although the proposed node selection for connecting the BC source in 94% of casesled to the calculation of all DC solutions, it is possible to carry out the analysis while connecting the BC source to other selected or even all nodes. The only limitation here are nodes corresponding to the gate(s) of MOS transistors only or the ungrounded terminal of an ideal voltage source. In these cases, the deflation method will not lead to the determination of subsequent solutions. Further research may involve supplementing the host application with procedures for testing the local stability of the equilibrium points corresponding to each operating point.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable for studies not involving humans or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: Batch files for the circuits under study and their schematics are available and can be shared with interested parties. The simulations use a commercial version of the IsSpice4 software, and the source code of the host application calls the software and generates netlists matched to this version of SPICE. As a result, the application is not shared. In order to use the proposed method, it is necessary to write an application that matches the available SPICE version or to use only the SPICE program in a GUI environment, as described in Section 3.1.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABM	Analog behavioral model
AC	Alternating current
CMOS	Complementary metal–oxide–semiconductor
DC	Direct current
EDA	Electronic design automation
MOS	Metal–oxide–semiconductor
MOSFET	Metal–oxide–semiconductor field-effect transistor
PWL	Piecewise linear
RAM	Random-access memory

References

1. Fortuna, L.; Buscarino, A. Nonlinear Technologies in Advanced Power Systems: Analysis and Control. *Energies* **2022**, *15*, 5167. [CrossRef]
2. Fortuna, L.; Buscarino, A. Chaos. *Appl. Sci.* **2022**, *12*, 11793. [CrossRef]
3. Crutchley, D.; Zwolinski, M. Globally convergent algorithms for DC operating point analysis of nonlinear circuits. *IEEE Trans. Evol. Comput.* **2003**, *7*, 2–10. [CrossRef]
4. Pastore, S. Fast and Efficient Search for All DC Solutions of PWL Circuits by Means of Oversized Polyhedra. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2009**, *56*, 2270–2279. [CrossRef]
5. Trajkovic, L. DC operating points of transistor circuits. *Nonlinear Theory Its Appl. IEICE* **2012**, *3*, 287–300. [CrossRef]
6. Ushida, A.; Yamagami, Y.; Nishio, Y.; Kinouchi, I.; Inoue, Y. An efficient algorithm for finding multiple DC solutions based on the SPICE-oriented Newton homotopy method. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2002**, *21*, 337–348. [CrossRef]
7. Fox, R.; Nagarajan, M. Multiple operating points in a CMOS log-domain filter. *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.* **1999**, *46*, 705–710. [CrossRef]
8. Goldgeisser, L.; Green, M. On the topology and number of operating points of MOSFET circuits. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2001**, *48*, 218–221. [CrossRef]

9. Shou, X.; Green, M. A family of CMOS latches with 3 stable operating points. In Proceedings of the ISCAS 2001, 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196), Sydney, NSW, Australia, 6–9 May 2001; Volume 1, pp. 109–112. [CrossRef]
10. Liu, Z.; Li, Y.; Geiger, R.L.; Chen, D. Auto-identification of positive feedback loops in multi-state vulnerable circuits. In Proceedings of the 2014 IEEE 32nd VLSI Test Symposium (VTS), Napa, CA, USA, 13–17 April 2014; pp. 1–5. [CrossRef]
11. Lee, J.; Chiang, H.D. Constructive homotopy methods for finding all or multiple DC operating points of nonlinear circuits and systems. In Proceedings of the 2000 IEEE International Symposium on Circuits and Systems (ISCAS), Geneva, Switzerland, 28–31 May 2000; Volume 4, pp. 525–528. [CrossRef]
12. ICAP4. *Working with Model Libraries*; Intusoft: San Pedro, CA, USA, 2000; Rev. 09/00.
13. IsSpice4. *Users Guides, Vol. 1,2, Rev 04/08*; Intusoft: Carson, CA, USA, 2008.
14. Yamamura, K.; Lai, L.; Kuroki, W. A SPICE-Oriented Method for Finding DC Operating Points of Nonlinear Circuits Containing Piecewise-Linear Macromodels. In Proceedings of the 2006 International Conference on Communications, Circuits and Systems, Guilin, China, 25–28 June 2006; Volume 4, pp. 2385–2389. [CrossRef]
15. Tadeusiewicz, M.; Hałgas, S. A method for finding multiple dc operating points of short channel CMOS circuits. *Circuits Syst. Signal Process.* **2013**, *32*, 2457–2468. [CrossRef]
16. Kolev, L. An interval method for global nonlinear analysis. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2000**, *47*, 675–683. [CrossRef]
17. Tadeusiewicz, M.; Hałgas, S. A method for the analysis of transistor circuits having multiple DC solutions. *AEU-Int. J. Electron. Commun.* **2006**, *60*, 582–589. [CrossRef]
18. Yamamura, K.; Suda, K. An Efficient and Practical Algorithm for Finding All DC Solutions of Nonlinear Circuits. In Proceedings of the 2007 International Conference on Communications, Circuits and Systems, Kokura, Japan, 11–13 July 2007; pp. 1111–1115. [CrossRef]
19. Uatrongjit, S. Finding All DC Solutions of Diode and BJT Circuits by Interval Method and Modified LP-Narrowing Technique. *IEEJ Trans. Electr. Electron. Eng.* **2010**, *5*, 621–626. [CrossRef]
20. Tadeusiewicz, M.; Hałgas, S. Analysis of transistor circuits having multiple DC solutions with the thermal constraint. *COMPEL - Int. J. Comput. Math. Electr. Electron. Eng.* **2011**, *30*, 1350–1362. [CrossRef]
21. Deng, J.; Batselier, K.; Zhang, Y.; Wong, N. An efficient two-level DC operating points finder for transistor circuits. In Proceedings of the 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 1–5 June 2014; pp. 1–6. [CrossRef]
22. Tadeusiewicz, M.; Hałgas, S. A contraction method for locating all the DC solutions of circuits containing bipolar transistors. *Circuits Syst. Signal Process.* **2012**, *31*, 1159–1166. [CrossRef]
23. Akhter, I.A.; Reiher, J.; Greenstreet, M.R. Finding All DC Operating Points Using Interval Arithmetic Based Verification Algorithms. In Proceedings of the 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), Florence, Italy, 25–29 March 2019; pp. 1595–1598. [CrossRef]
24. Uatrongjit, S.; Kaewkham-Ai, B.; Prakobwaitayakitt, K. Finding All DC Operating Points of Nonlinear Circuits Based on Interval Linearization and Coordinate Transformation. In Proceedings of the 2022 International Electrical Engineering Congress (iEECON), Khon Kaen, Thailand, 9–11 March 2022; pp. 1–4. [CrossRef]
25. Somogyi, A.J. Analytical properties of mos transistor characteristics. *Int. J. Circuit Theory Appl.* **1982**, *10*, 167–171. [CrossRef]
26. Tadeusiewicz, M. Global and local stability of circuits containing MOS transistors. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2001**, *48*, 957–966. [CrossRef]
27. Goldgeisser, L.; Green, M. A method for automatically finding multiple operating points in nonlinear circuits. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2005**, *52*, 776–784. [CrossRef]
28. Gajani, G.S.; Brambilla, A.; Premoli, A. Numerical Determination of Possible Multiple DC Solutions of Nonlinear Circuits. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2008**, *55*, 1074–1083. [CrossRef]
29. Lee, J.; Chiang, H.D. Constructive homotopy methods for finding all or multiple DC operating points of nonlinear circuits and systems. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2001**, *48*, 35–50. [CrossRef]
30. Tadeusiewicz, M.; Hałgas, S. Computing multivalued input-output characteristics in the circuits containing bipolar transistors. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2004**, *51*, 1859–1867. [CrossRef]
31. Mejia, C.H.; Reyes, A.S.; Vazquez-Leal, H. Existence of multiple operating points in memristive circuits. In Proceedings of the 2012 IEEE 3rd Latin American Symposium on Circuits and Systems (LASCAS), Playa del Carmen, Mexico, 29 February–2 March 2012; pp. 1–4. [CrossRef]
32. Tadeusiewicz, M.; Hałgas, S. Numerical analysis of direct current circuits containing bipolar and metal oxide semiconductor transistors. *Int. J. Numer. Model. Electron. Netw. Devices Fields* **2014**, *27*, 935–948. [CrossRef]
33. Weber, H.; Trajkovic, L.; Mathis, W. Finding DC Operating Points of Nonlinear Circuits Using Carleman Linearization. In Proceedings of the 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), Lansing, MI, USA, 9–11 August 2021; pp. 1078–1081. [CrossRef]
34. Tadeusiewicz, M.; Hałgas, S. Analysis of BJT circuits having multiple DC solutions using deflation technique. In Proceedings of the 2012 International Conference on Signals and Electronic Systems (ICSES), Wroclaw, Poland, 18–21 September 2012; pp. 1–4. [CrossRef]

35. Liu, Z.; Li, Y.; Duan, Y.; Geiger, R.L.; Chen, D. Identification and break of positive feedback loops in Trojan States Vulnerable Circuits. In Proceedings of the 2014 IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne, VIC, Australia, 1–5 June 2014; pp. 289–292. [CrossRef]
36. Li, Y.; Liu, Z.; Chen, D. Efficient Verification Against Undesired Operating Points for MOS Analog Circuits. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2017**, *64*, 2134–2145. [CrossRef]
37. Brown, K.M.; Gearhart, W.B. Deflation techniques for the calculation of further solutions of a nonlinear system. *Numer. Math.* **1971**, *16*, 334–342. [CrossRef]
38. Farrell, P.E.; Birkisson, A.; Funke, S.W. Deflation Techniques for Finding Distinct Solutions of Nonlinear Partial Differential Equations. *SIAM J. Sci. Comput.* **2015**, *37*, A2026–A2045. [CrossRef]
39. Ojika, T.; Watanabe, S.; Mitsui, T. Deflation algorithm for the multiple roots of a system of nonlinear equations. *J. Math. Anal. Appl.* **1983**, *96*, 463–479. [CrossRef]
40. Huang, T.M.; Lin, W.W.; Mehrmann, V. A Newton-Type Method with Nonequivalence Deflation for Nonlinear Eigenvalue Problems Arising in Photonic Crystal Modeling. *SIAM J. Sci. Comput.* **2016**, *38*, B191–B218. [CrossRef]
41. Luo, X.l.; Xiao, H. Continuation Newton methods with deflation techniques and quasi-genetic evolution for global optimization problems. *arXiv* **2021**, arXiv:2107.13864.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.