SIMPLE IMPLEMENTATIONS OF HOMOTOPY ALGORITHMS FOR FINDING DC SOLUTIONS OF NONLINEAR CIRCUITS

A. Dyess †
ECE Department
University of Alabama
Tuscaloosa, AL 35487
ad3160@labs.eng.ua.edu

E. Chan, H. Hofmann, W. Horia
EECS Department
University of California
Berkeley, CA 94720-1770
chan1, heath, wjwh
@cory.eecs.berkeley.edu

Lj. Trajković *
School of Engineering Science
Simon Fraser University
Burnaby, BC, V5A 1S6, Canada
ljilja@cs.sfu.ca

ABSTRACT

We describe simple software implementations of parameter embedding (also called continuation and homotopy) algorithms for calculating dc operating points of nonlinear circuits. Past implementation of homotopy algorithms in industrial circuit simulators proved that they were viable options to resolving convergence difficulties when finding circuits' dc operating points. These software implementations involved proprietary circuit simulation tools and sophisticated software implementation of homotopy algorithms. The implementation described here, relies on commercially available MATLAB tools. In spite of its simplicity, our implementation proved powerful enough to solve benchmark nonlinear circuits with multiple dc operating points.

1. INTRODUCTION

Parameter embedding methods are robust and accurate numerical techniques for solving nonlinear algebraic equations [1], [2]. They can be used to find multiple solutions of equations that possess more than one solution [3]. A class of embedding algorithms called probability-one homotopy algorithms that promise global convergence [4] have been implemented in a publicly available software package HOMPACK [5]. Past research and implementations of homotopy algorithms for finding circuit dc operating points indicated promising results [6]-[11]. These algorithms have been used to find solutions to highly nonlinear circuits that could not be simulated [12] using conventional numerical methods. They are also useful in finding dc operating points of multistable circuits. The main drawback in using homotopy methods is their computational intensity. Therefore, they are most suitable for solving difficult nonlinear problems where initial solutions are hard to estimate or multiple solutions are desired. For circuits that fall in this category, homotopy algorithms offer a very attractive alternative.

This work was supported by the †Computing Research Association Distributed Mentor Project, and the *National Science Foundation under Grant GER-9550153.

We describe here simple software implementations of homotopy algorithms using the MATLAB software package [13]. In spite of it simplicity, our implementation proved powerful enough to solve benchmark nonlinear circuits with multiple operating points.

2. HOMOTOPY METHODS: BACKGROUND

Homotopy methods are used to solve systems of nonlinear algebraic equations and can be applied to a large variety of problems. We are most interested in solving the zero finding problem

$$\mathcal{F}(\mathbf{x}) = \mathbf{0}, \qquad (1)$$

where $\mathbf{x} \in \mathcal{R}^n$, $\mathcal{F} : \mathcal{R}^n \rightarrow \mathcal{R}^n$. (Note that the fixed point problem can be easily reformulated as a zero finding problem.)

We create the homotopy function $\mathcal{H}(\mathbf{x}, \lambda)$ by embedding a parameter λ into $\mathcal{F}(\mathbf{x})$ and thus obtaining an equation of higher dimension

$$\mathcal{H}(\mathbf{x}, \lambda) = \mathbf{0}, \qquad (2)$$

where $\lambda \in \mathcal{R}$, $\mathcal{H} : \mathcal{R}^n \times \mathcal{R} \rightarrow \mathcal{R}^n$. For $\lambda = 0$,

$$\mathcal{H}(\mathbf{x}, 0) = \mathbf{0} \qquad (3)$$

is an easy equation to solve, and for $\lambda = 1$,

$$\mathcal{H}(\mathbf{x}, 1) = \mathbf{0} \qquad (4)$$

is the original problem (1). The parameter λ is called the continuation or homotopy parameter.

An example of a homotopy is

$$\mathcal{H}(\mathbf{x}, \lambda) = (1 - \lambda)\mathcal{G}(\mathbf{x}) + \lambda\mathcal{F}(\mathbf{x}). \qquad (5)$$

Hence, $\mathcal{H}(\mathbf{x}, 0) := \mathcal{G}(\mathbf{x}) = \mathbf{0}$ has an easy solution, while $\mathcal{H}(\mathbf{x}, 1) := \mathcal{F}(\mathbf{x}) = \mathbf{0}$ is our original problem. By following solutions of $\mathcal{H}(\mathbf{x}, \lambda) = \mathbf{0}$ as λ varies from 0 to 1, we reach the solution to $\mathcal{F}(\mathbf{x}) = \mathbf{0}$.

The solutions trace a path known as the zero curve. Various numerical problems may occur depending on the behavior of this curve. One problem occurs if the curve folds back. At the turning point the values of λ decrease as the path progresses. Increasing λ from 0 to 1 results in "losing" the curve. The difficulty is resolved by making λ a function of a new parameter: the arc length s. This method is known as the arc length continuation [2], [5].

3. DC OPERATING POINT ANALYSIS

We used homotopy methods to find the dc operating points of nonlinear circuits. This method of solving equations was implemented through the use of two software programs. First, a parser generates the mathematical equations from a description of the circuit in the commonly used SPICE format [14], [15]. These equations are then solved by a MATLAB tool [13]. We tested the method by solving Chua's four-transistor benchmark circuit [16]. As expected, nine distinct solutions were found.

3.1. Parser

Implementation of the homotopy method requires that the set of equations that describe the circuit be specified. Only for very simple circuits, these equations can be written by hand. The parser is a C++ program that accepts a SPICE input file [14], [15], and produces either nodal analysis or modified nodal analysis [17] circuit equations, as well as their Jacobian matrices.

We used simple models of nonlinear circuits components (diodes and bipolar junction transistors) to demonstrate the parser correctness. More realistic models could be implemented by specifying the nonlinear equations that govern the components' behavior.

3.2. Solver

There are several approaches to implement homotopy methods [2]. We opted for algorithms based on the ordinary differential equations.

The solution of the equation

$$\mathcal{H}(\mathbf{x}(s), \lambda(s)) = \mathbf{0}, \tag{6}$$

where s is the *arc length* parameter, is a trajectory

$$\mathbf{y}(s) = \left(\begin{array}{c} \lambda(s) \\ \mathbf{x}(s) \end{array} \right). \tag{7}$$

This trajectory is found by solving the differential equation

$$\frac{d}{ds}\mathcal{H}(\mathbf{x}(s), \lambda(s)) = \mathbf{0}, \tag{8}$$

with conditions

$$\lambda(0) = 0, \quad \mathbf{x}(0) = \mathbf{a}, \quad \text{and} \quad \| \frac{d\lambda}{ds}, \frac{d\mathbf{x}}{ds} \|_2 = 1. \tag{9}$$

Differential equation (8) can be written as

$$\mathbf{P}(\mathbf{y})\dot{\mathbf{y}} := \left[\frac{\partial \mathcal{H}}{\partial \lambda} \frac{\partial \mathcal{H}}{\partial \mathbf{x}} \right] \left[\begin{array}{cc} \frac{d\lambda}{ds} & \frac{d\mathbf{x}}{ds} \end{array} \right]. \tag{10}$$

We wish to solve

$$\mathbf{P}(\mathbf{y})\dot{\mathbf{y}} = \mathbf{0} \tag{11}$$

for $\dot{\mathbf{y}}$. The solution is unique if the extended Jacobian matrix (8) is of full rank. Conditions (9) define the starting point for $\mathbf{x}$, the starting value of λ, and ensure that the sign and the magnitude of $\dot{\mathbf{y}}$ are fixed in the implementation. The solution $\dot{\mathbf{y}}$ is found by solving linear differential equation (11) using standard linear solvers via the QR factorization algorithm [18].

Once the derivatives are determined, we used the variable-step predictor-corrector method to find $\mathbf{y}(s)$ from its derivative that were found in the previous step. The method proved superior to the Runge-Kutta methods that we initially used.

Finally, the "end game" was used to determine the step size so that the solution to $\mathbf{y}(s)$ for $\lambda = 1$ can be reached. A cubic spline interpolation of $\lambda(s)$ and a solution to $\lambda(s) = 1$ (the smallest root that is greater than the current value of s) were used to predict the next step size. Once λ is within the preselected tolerance, the value of $\mathbf{x}$ was assumed to be the sought solution.

3.3. Example

Nine dc operating points of the four-transistor circuit [16] shown in Figure 1 were found by using our MATLAB implementation of the homotopy algorithm. Parser was used to generate modified nodal analysis equations, and the simple homotopy function An example of a homotopy is

$$\mathcal{H}(\mathbf{x}, \lambda) = (1 - \lambda)\mathbf{G}(\mathbf{x} - \mathbf{a}) + \lambda\mathcal{F}(\mathbf{x}), \tag{12}$$

where $\mathbf{G}$ is a diagonal scaling matrix, and $\mathbf{a}$ is a starting vector.

MATLAB can also generate plots of the homotopy paths for the unknown node voltages and for the currents flowing through each voltage source. They are shown in Figure 2 and Figure 4, respectively. By zooming in on the path for an individual node voltage and current, we can see that each path crosses the vertical line $\lambda = 1$ nine times. These paths are shown in Figures 3 and 5.

The results from MATLAB can be compared with solutions from other homotopy methods [11]. Even though Newton-Raphson method solvers like PSPICE and SPICE 3F5 will calculate only one dc operating point, it is possible to give PSPICE an initial guess that is close to a desired solution by using the .NODESET option. In this manner, by using the MATLAB results as a starting point, we have found all nine dc operating points.

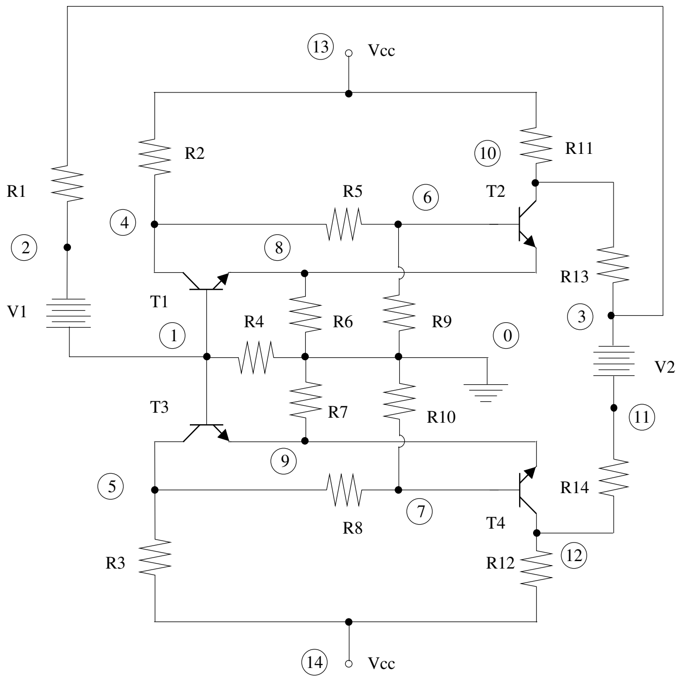

Figure 1: Four-transistor circuit that has nine dc operating points. Circuit parameters: R1 = 1KΩ, R2 = R3 = 4KΩ, R4 = 5KΩ, R5 = R8 = 30KΩ, R6 = R7 = 0.5KΩ, R9 = R10 = 10.1KΩ, R11 = R12 = 4KΩ, R13 = R14 = 30KΩ, V1 = 10V, V2 = 2V, and VCC = 12V.

4. CONCLUDING REMARKS

Our implementation, which employs commercially widely available software package MATLAB, illustrates that implementations of homotopy algorithms need not necessarily rely on large numerical solvers or proprietary circuit simulation tools. Furthermore, simple homotopy functions proved adequate for solving some difficult benchmark circuits. We successfully used the current implementation to find nine dc operating points of a benchmark four-transistor circuit. The accuracy of the results was verified by comparison with PSPICE solutions and results of other homotopy implementations.

Our implementation could be made more efficient. It takes several minutes to calculate the nine solutions and produce the resulting plot for our example circuit. The speed of the algorithm and the number of solutions found depend on the starting point of the homotopy path. One possible improvement is to use an algorithm to determine a better starting point, rather than choosing a random value. The algorithm could also be improved to eliminate numerical instability for values of λ very close to 1 (the so called "end game" [19]). Other homotopy mappings could also be implemented.

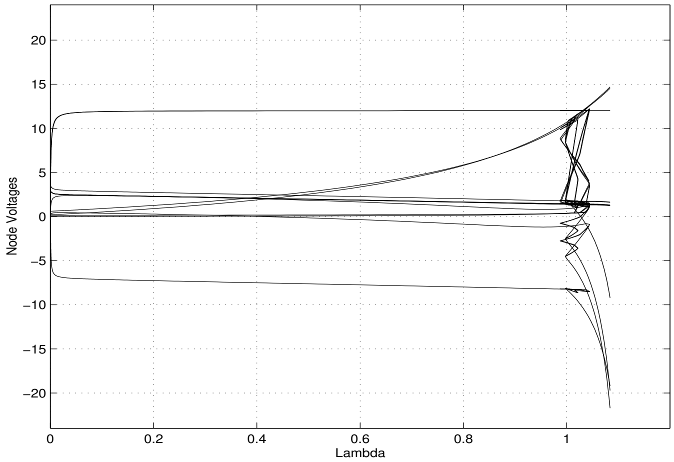

Figure 2: Homotopy paths for the fourteen node voltages of the four-transistor circuit.

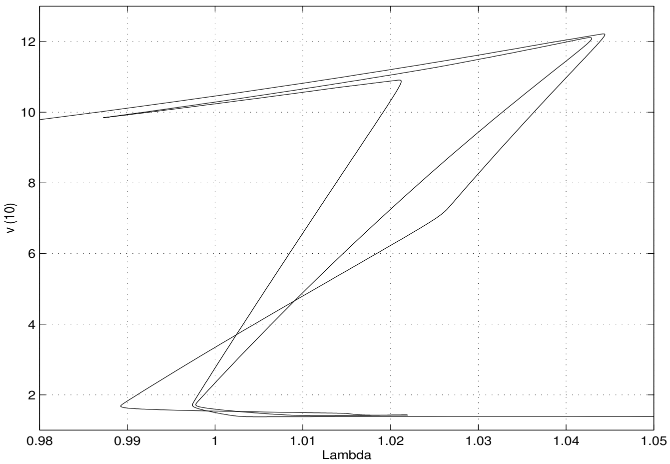

Figure 3: Closer view of the homotopy path for the voltage at node 10.

5. REFERENCES

[1] E. L. Allgower and K. Georg, *Numerical Continuation Methods: An Introduction.* New York: Springer-Verlag Series in Computational Mathematics, 1990, pp. 1–15.

[2] L. T. Watson, "Globally convergent homotopy algorithms for nonlinear systems of equations," *Nonlinear Dynamics,* vol. 1, pp. 143–191, Feb. 1990.

[3] W. Rheinboldt and J. V. Burkardt, "A locally parameterized continuation process," *ACM Transactions on Mathematical Software,* vol. 9, no. 2, pp. 215–235, June 1983.

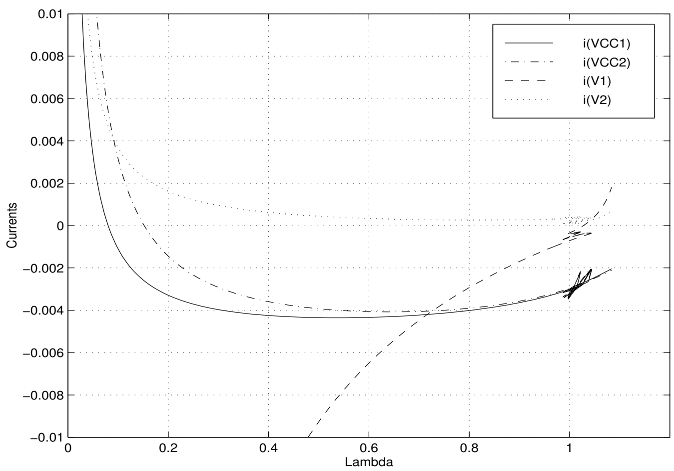

Figure 4: Homotopy paths for the four currents flowing through the voltage sources of the four-transistor circuit.

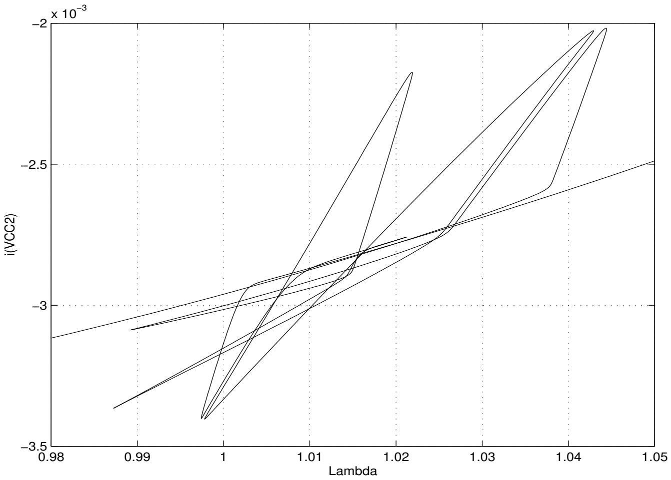

Figure 5: Closer view of the homotopy path for the current flowing through the voltage source connected to node 14.

[4] S. Chow, J. Mallet-Paret, and J. A. Yorke, "Finding zeroes of maps: homotopy methods that are constructive with probability one," *Mathematics of Computation,* vol. 32, no. 143, pp. 887–899, July 1978.

[5] L. T. Watson, S. C. Billups and A. P. Morgan, "Algorithm 652: HOMPACK: A suite of codes for globally convergent homotopy algorithms," *ACM Trans. Mathematical Software*, vol. 13, no. 3, pp. 281–310, Sept. 1987.

[6] L. Goldgeisser and M. M. Green, "A novel algorithm that finds multiple operating points of nonlinear circuits automatically, *Proc. ISCAS '98*, Monterey, CA, June 1998, TPA 12-3.

[7] R. C. Melville, Lj. Trajković, S. C. Fang and L. T. Watson, "Artificial parameter homotopy methods for the dc operating point problem," *IEEE Trans. on CAD,* vol. 12, no. 6, pp. 861–877, June 1993.

[8] Lj. Trajković and W. Mathis, "Parameter embedding methods for finding dc operating points: Formulation and implementation," *Proc. NOLTA '95*, Las Vegas, NV, Dec. 1995, pp. 1159–1164.

[9] Lj. Trajković, E. Fung, and S. Sanders, "HomSPICE: Simulator with homotopy algorithms for finding Parameter embedding methods for finding dc and steady-state solutions of nonlinear circuits," *Proc. ISCAS '98*, Monterey, CA, June 1998, TPA 10-2.

[10] D. Wolf and S. Sanders, "Multiparameter homotopy methods for finding DC operating points of nonlinear circuits," *IEEE Trans. Circuits Syst.,* vol. 43, pp. 824–838, Oct. 1996.

[11] K. Yamamura, T. Sekiguchi, and Y. Inoue, "A globally convergent algorithm using the fixed-point homotopy for solving modified nodal equations," *Proc. NOLTA '96*, Kochi, Japan, pp. 463–466, Oct. 1996.

[12] R. Kielkowski, *Inside SPICE*, Second edition. New York, NY: McGraw-Hill, 1998.

[13] D. Hanselman and B. Littlefield, *Mastering MATLAB: A comprehensive tutorial and reference*. Upper Saddle River, NJ: Prentice Hall, 1996.

[14] T. L. Quarles, A. R. Newton, D. O. Pederson, and A. Sangiovanni-Vincentelli, "SPICE 3 Version 3F5 User's Manual," Department of EECS, University of California, Berkeley, March 1994.

[15] A. Vladimirescu, *The SPICE Book*. New York: John Wiley & Sons, Inc., 1994.

[16] L. O. Chua and A. Ushida, "A switching-parameter algorithm for finding multiple solutions of nonlinear resistive circuits," *Int. J. Circuit Theory Appl.*, vol. 4, pp. 215–239, 1976.

[17] C. W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 504–509, Jan. 1975.

[18] J. D. Lambert, *Numerical methods for ordinary differential equations: The initial value problem*. New York, NY: John Wiley, 1991.

[19] M. Sosokina, L. T. Watson, and D. E. Stewart, "Note on the end game in homotopy zero curve tracking," *ACM Trans. Mathematical Software*, vol. 22, no. 3, pp. 281–287, Sept., 1996.