Solution of implicit equations in SPICE

Stav Zaitsev
Stav.Zaitsev@praxisdynamics.com.au
Praxis Dynamics, Seaford Meadows, South Australia

4 July 2012

Keywords: Implicit equation, SPICE, feedback, MEMS, pull-in

1 Introduction

Implicit functions connecting two (or more) different variables are frequently encountered in mathematical modeling of different physical systems. Examples include the magnetization hysteresis curve $B-H$, memory behaviour of memristors, the self-consistent charge transport in materials under electron beam irradiation, the steady state behaviour of a nonlinear micromechanical resonator, and a wide range of other self-consistent problems and nonlinear processes.

An implicit functional relation can be written as

$$f(x,y)=0, \tag{1}$$

where x and y are some model variables. Here, we will assume that y is externally controlled (e.g., excitation force, input voltage, etc.), while x is the "slave" variable, whose value is determined by Eq. (1). If this equation can be rearranged as

$$x=g(y),$$

the task of determining x becomes trivial. However, in many cases, the transition $f \to g$ is not possible (e.g., f is a polynomial function of fifth order or higher) or not practical (e.g., f is a cubic or quartic polynomial function). Moreover, Eq. (1) may have several solutions for the given value of the independent variable, and the specific solution realized in the system depends on the history of its evolution.

When trying to simulate such systems, the go-to solution usually includes some sort of an advanced numerical analysis software, such as Matlab, or a full-blown multiphysics simulator, such as COMSOL. Although these tools are well suited for the task, the integration of their results with SPICE simulations can be complicated, and the flexibility of parameter tuning, and exploring different what-if scenarios and circuit topologies in a SPICE simulator is hindered significantly.

Here, we show a simple way to solve equations similar to Eq. (1) directly in SPICE time domain simulations.

2 Using feedback to solve implicit equation

The idea of solving an implicit equation by using a high gain feedback loop dates back to the days of analogue computers. Consider the feedback loop shown in Fig. 1. At a steady state, the

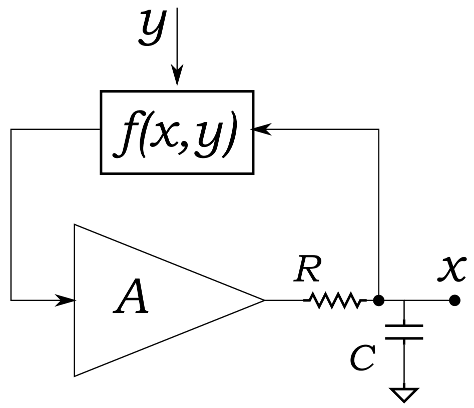

Figure 1: Feedback loop for approximate solution of an implicit equation $f(x,y) = 0$ [see Eq. (1)], where y is the independent variable, and x is the dependent variable.

following relation is maintained:

$$f(x,y) = \frac{x}{A} \xrightarrow{|A| \gg x} 0, \tag{2}$$

where A is the gain of the amplifier in Fig. 1. Thus, the output x is an approximate solution of Eq. (1).

The subset of solutions of Eq. (1) that can be found by the system shown in Fig. 1 depends on the sign of the amplifier gain A. In an example shown in Fig. 2, the system will converge to $x \approx x_3$ for positive A, and to $x \approx x_2$ for negative A. In other words, the solutions $x \approx x_1$ and $x \approx x_3$ are stable for $A > 0$, while the solution $x \approx x_2$ is not stable. The opposite is correct for the case $A < 0$. Therefore, it is possible to ensure convergence to physically stable solutions by the correct choice of the sign of the feedback gain.

The steady state error in x can be estimated as

$$\left|\frac{\delta x_i}{x_i}\right| \approx \frac{1}{\left|A\frac{\partial f(x_i,y)}{\partial x}\right|}, \tag{3}$$

where δx_i is the difference between the true solution x_i and the system's output x. In the important case in which $\partial f(x_i,y)/\partial x = 0$, which usually corresponds to a loss of a stable solution due to a saddle-node bifurcation (see the example in the next Section), the steady state error can be shown to be

$$\left|\frac{\delta x_i}{x_i}\right| \approx \sqrt{\frac{2}{\left|Ax_i\frac{\partial^2 f(x_i,y)}{\partial x^2}\right|}}. \tag{4}$$

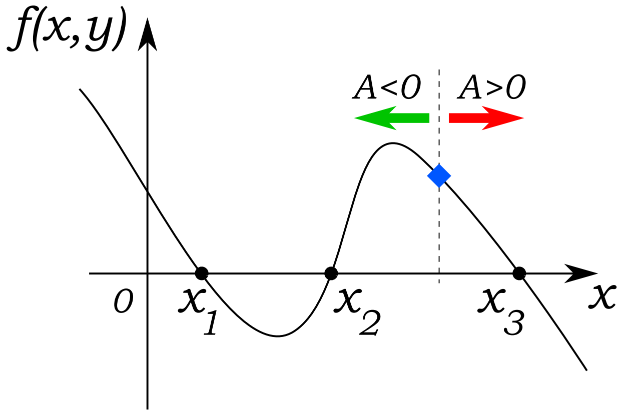

Figure 2: The impact of the sign of the amplifier gain A on the specific solution to which the system shown in Fig. 1 converges. Possible solutions of Eq. (1) are denoted x_i. The initial condition is denoted by a blue diamond and a vertical dashed line. The red (green) arrow shows the direction to the closest solution to which the system will converge for $A > 0$ ($A < 0$), i.e., x_3 (x_2).

It follows from the above discussion that large values of the gain A are required to achieve reasonable accuracy of the solutions. However, due to the discreet nature of the time domain SPICE simulations, excessively large gain can cause the system to skip a solution, or can result in simulator convergence issues. As usual, a trade-off exists, in this case between the accuracy and the response time of the system, set by the low pass RC output filter (see Fig. 1). Assuming that the simulation time step δt is much smaller than the filter time constant RC, the change in the output signal δx can be estimated as

$$\delta x = (Af - x)\,\frac{\delta t}{RC}. \tag{5}$$

If the system output is sufficiently far from the solution, i.e., $Af \gg x$, the following approximate condition can be formulated for the maximum allowable time step size:

$$\delta t < \frac{\Delta_x}{A \max|f|} RC, \tag{6}$$

where Δ_x is the distance between two successive solutions, and $\max|f|$ is the maximum value of $f(x,y)$ in the interval between these solutions. Fortunately, such small steps are usually required only during the initial settling down period (or following large changes in y). The majority of modern simulators use a variable time step size, allowing accurate convergence to the steady state without significant penalty to the total simulation time and the number of points.

Finally, it must be emphasized that in order to maintain a reliable solution at all times, the output RC filter should be fast enough to follow the variation of $f(x,y)$ due to the changes of the independent variable y.

3 Example: static pull-in in MEMS

An interesting example that requires a continuing solution of an equation similar to Eq. (1) comes from the field of microelectromechanical systems (MEMS). Consider a simple MEMS capacitor shown in Fig. 3. This model can describe a wide range of electrostatically actuated

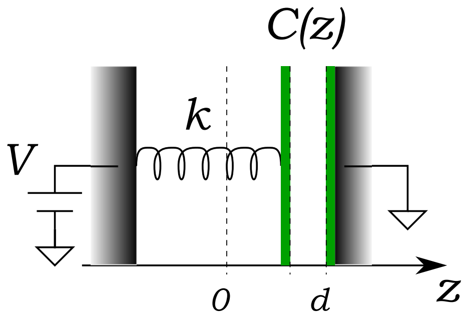

Figure 3: MEMS parallel plates capacitor. The left plate is suspended on a micromechanical spring with a spring coefficient k, and its displacement is denoted z. The right plate is static and grounded. The capacitance between the plates is $C(z) = C(z=0)/(1-z/d)$. A constant voltage V is applied between the plates.

devices, including comb drives, RF MEMS varactors and switches, etc.

Here, we assume that the micromechanical system is over-damped, and, in addition, that the changes in the actuating voltage V are slow, i.e., the system is quasi-static. The position of the suspended capacitor plate is determined by the equilibrium between the mechanical elastic force $-kz$ and the electrostatic attractive force $V^2C_0/2d(1-z/d)^2$, where $C_0 = C(z=0)$. It follows that the position of the plate, z, is a solution of the following equation:

$$x - \frac{V^2C_0}{2kd^2(1-x)^2} = 0, \tag{7}$$

$$K = \frac{C_0}{2kd^2}, \tag{8}$$

where we have defined an non-dimensional displacement $x = z/d$ for convenience.

In general, Eq. (7) is a cubic equation and, therefore, has three solutions. Analytical expressions exist for these solutions, but they are extremely cumbersome, and do not provide the user with any insight about the system's stability. It is, therefore, more practical to solve this equation using the feedback technique described above, especially if controlling and driving electronics is simulated as well.

One of the solutions of Eq. (7) is always real but larger than 1 ($z > d$), i.e., it is not physical. A real physical solution exists only for voltages below some critical value V_{pi}, at which the displacement reaches its maximum stable value $x = \frac{1}{3}$. Increasing the voltage further will result in a loss of mechanical stability, and the suspended capacitor plate will snap toward the static plate. This phenomenon is known as pull-in, and can be quite destructive and even irreversible in various MEMS devices.

The simulation circuit is presented in Fig. 4, where a behavioural source `B` is employed to calculate the left side of Eq. (7) and to provide the gain A. The following typical values are used for different parameters: $d = 10\,\mu\text{m}$, $C_0 = 200\,\text{fF}$, $k = 1\,\text{N}/\text{m}$, and the gain is $A = -1 \times 10^5$.

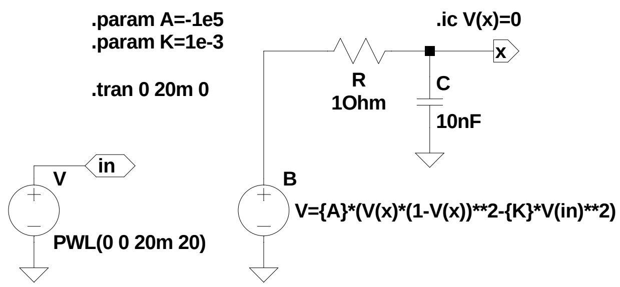

Figure 4: A simulation circuit designed to solve the equation $x(1-x)^2 - KV^2 = 0$, where $K = C_0/2kd^2$ [see Eq. (7)]. The parameter values are: $d = 10\,\mu\text{m}$, $C_0 = 200\,\text{fF}$, $k = 1\,\text{N}/\text{m}$, $K = 1 \times 10^{-3}\,1/\text{V}^2$, and the gain $A = -1 \times 10^5$. The output RC filter time constant is $10\,\text{ns}$. The excitation voltage V_{in} is swept from $0\,\text{V}$ to $20\,\text{V}$ at a rate of $1\,\text{V}/\text{ms}$.

The system is expected to lose stability at $x = \frac{1}{3}$, at which point

$$V_{\text{pi}} = \frac{2}{3}\sqrt{\frac{1}{3K}}, \tag{9}$$

Using the parameter values given above, the expected pull-in voltage is $V_{\text{pi}} = 12.17\,\text{V}$. In the simulation, the excitation voltage V_{in} is swept from $0\,\text{V}$ to $20\,\text{V}$ at a rate of $1\,\text{V}/\text{ms}$.

The results are shown in Fig. 5. As expected, three real solutions exists, two of them physical. The lowest one is stable and is followed by the system starting from zero excitation voltage. The middle one is unstable and is located on the separatrix, i.e., for any initial conditions above this solution, the system will undergo a pull-in, even if a stable lower solution exists. When the excitation voltage exceeds the critical value V_{pi}, these two physical solutions are eliminated in a saddle-node bifurcation, and only the third (upper) stable solution remains. This solution is not physical, because $x > 1$, and is shown here for completeness. In a real simulation, the output of the system would be limited to $0 \leq x \leq 1$.

It remains to show that the worst case error estimation given in Eq. (4) is valid. Indeed, a quick calculation shows that at the bifurcation point ($V_{\text{in}} = V_{\text{pi}}$, $x = \frac{1}{3}$), the estimated relative error is $|\delta x/x| \approx 5 \times 10^{-3}$, while comparison of the exact solution with the simulation results gives $|\delta x/x| \approx 1 \times 10^{-3}$.

4 Summary

Using a simple feedback topology, it is possible to solve implicit equations directly in SPICE simulators during time domain simulations. This approach allows the engineer to simulate a variety of complex physical models directly in SPICE, models that may otherwise require specialized numerical software and laborious post-simulation integration of the results. In the example discussed here, the micromechanical pull-in effect is simulated with very high accuracy.

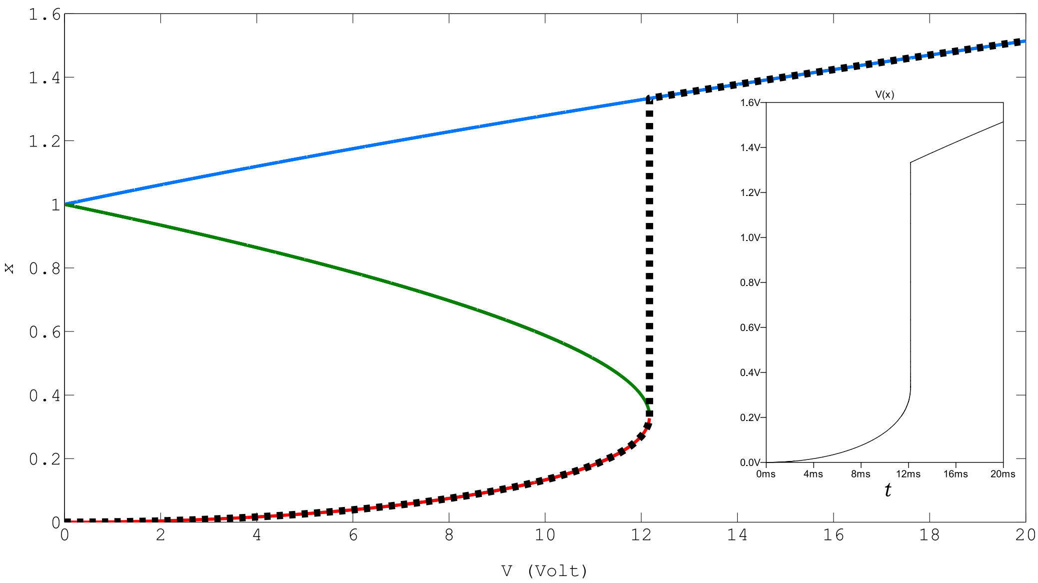

Figure 5: Comparison of the simulation results (black dotted line) with exact solutions of Eq. (7) (solid lines). Three real solutions exist. For low excitation voltages $V_{\text{in}} < V_{\text{pi}} = 12.17\,\text{V}$, the simulated solution coincides with the stable physical solution (red solid line). Above the pull-in voltage, the simulated solution jumps to the second stable solution (blue solid line). This solution is not physical. The remaining, unstable solution is denoted by a green solid line. In the inset, the actual time trace from the SPICE simulation (see Fig. 4) is shown.

It is shown, however, that a good understanding of the underlying model is needed, and careful choice of simulation parameters based on preliminary analytical analysis is required to ensure that the simulated system converges to a physical and stable solution if one exists, and deals correctly with cases in which such solution is not present.

All circuits and simulations presented in this paper were created using the LTspice IV simulator developed by Linear Technology Corporation.