The Relaxation-based Method with Fast Automatic Differentiation for Circuit Simulation and Analysis

Ganesh Kumar BASNET, Hisashi AOMORI and Mamoru TANAKA
Sophia University
Department of Electrical and Electronics Engineering
7-1, Kioi-cho, Chiyoda-ku, Tokyo 102-8554
JAPAN

Abstract: In this paper, we propose a relaxation-based method with Fast Automatic Differentiation (FAD) for solving nonlinear problems. We called this method as R-FAD in this paper. By the proposed R-FAD the large circuits of neurons having complex nonlinear and Piece-Wise Linear (PWL) functions are modeled in terms of macro-models and can be analyzed by relaxation-based algorithm keeping the structure of Jacobian matrix. In this method, the differential elements of the complex nonlinear function $f(x)$ are computed by the FAD and added to the Jacobian matrix. In simulation, we had interesting synchronization phenomena.

Key–Words: Relaxation, Fast Automatic Differentiation, Nonlinear Function, Piece-wise Linear Function, Cellular Neural Network

1 Introduction

The most important points in R-FAD are that it has parallelism, connective representation for neuron macro-model and capacity to analyze any nonlinearity of each neuron without user's program. The Gauss Seidel Method (GSM) which is a kind of relaxation-based algorithm has been used in R-FAD because the basic computation of each neuron processor in a set of neural parallel processors can be done by sum of products for weights and state voltages transferred from neighboring neuron processors in each iteration. The computation can be realized by virtual parallel computers (brain computer) architecture or by many threads in software.

The SPICE is general to use for analyzing transistor circuits [1],[2]. And the fundamental of SPICE is Newton Raphson Method (NRM) with LU decomposition. Different than the SPICE the R-FAD contains FAD for the analysis of nonlinearity as well as in R-FAD the user defines only the description of many nonlinear functions as their equations in the input file without program. Due to this application it is easier to analyze neurons with nonlinear functions in R-FAD. The R-FAD models and analyzes the neural networks having any complex nonlinear function $i = f(v)$ for each neuron state voltage v.

The FAD is a chain ruled-based technique to compute partial derivative value of a composite function [3],[4]. The FAD makes to compute the element of Jacobean matrix automatically.

In our proposed method, the relaxation method is used to solve the linearized equation for NRM. It is very important that a virtual capacitor between each node and the reference node is inserted to guarantee the diagonal dominant of Jacobian matrix for its convergence because the "Relaxation Method" can't be applied without the diagonal dominant of Jacobian Matrix [5],[6]. In simulation, we have shown the interesting phenomena for neural networks by our method.

2 Relaxation Method

We call the virtual relaxation method applied in this paper as Sophia Relaxation Program (SRP). Though the SRP has been used to analyze MOS and Bipolar transistor circuits [5] without the FAD as SPICE, we consider here the real application of the SRP is to analyze neural networks for parallelism and for nonlinearity by adding the FAD. It is important that any neuron described by nonlinear equation with respect to capacitive voltage can be analyzed based on numerical stability for nonlinear stiffness and keeping structure of Jacobian matrix for parallelism. The stiff nonlinear state equation is given by

$$\frac{d\mathbf{v}}{dt} = -\mathbf{F}(\mathbf{v}, \mathbf{u}, t) \qquad (1)$$

where $\mathbf{v}$ and $\mathbf{u}$ are node and input voltage vectors respectively. The function $\mathbf{F}$ represents the sum of cur-

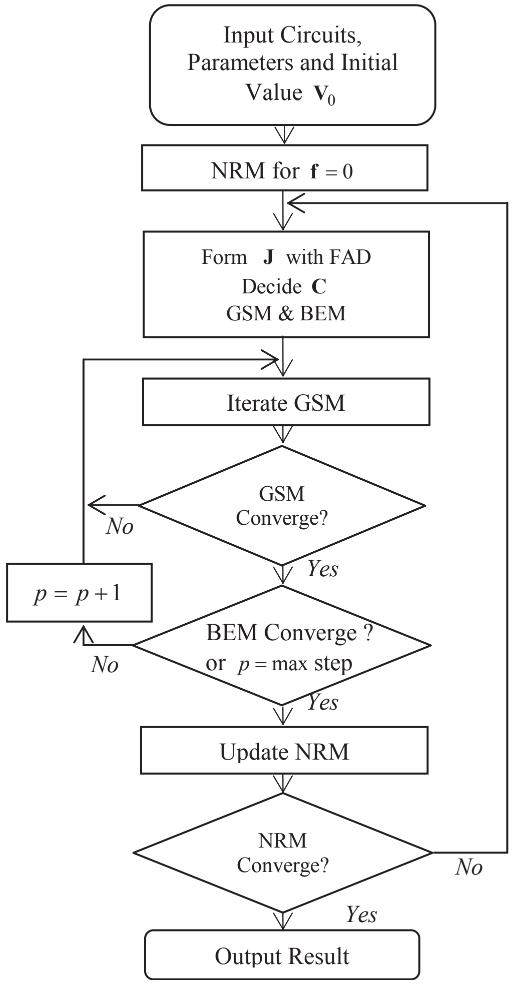

Figure 1: Flowchart of SRP.

rents flowing out from each node to incident conductive elements. The equation (1) is converted to nonlinear nodal algebraic equations by Backward Euler Method (BEM) in each real time step Δt as

$$\mathbf{f} = \frac{1}{\Delta t}(\mathbf{v}^{(m+1)} - \mathbf{v}^{(m)}) + \mathbf{F}^{(m+1)} \tag{2}$$

where superscript $(m+1)$ and (m) are discrete times. Let $\mathbf{J}$ be a Jacobian matrix by FAD as

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{v}}\Big|_{\mathbf{v}=\mathbf{v}_k^m} \tag{3}$$

then, NRM for $\mathbf{f} = 0$ is

$$\mathbf{J}\cdot\tilde{\mathbf{v}}_k^{m+1} = \mathbf{f}_k \tag{4}$$

where $\tilde{\mathbf{v}}_k^{m+1} = \mathbf{v}_k^{m+1} - \mathbf{v}_{k+1}^{m+1}$ for k, the step of NRM. The convergence criterion for solving the linearized equation by relaxation method must depend on whether $\mathbf{J}$ is diagonal dominant.

It is possible that the linearized equation (4) is assumed to be a linear virtual circuit. To make relaxation methods applicable for analysis of the virtual circuit, each adaptive virtual capacitor is inserted from less diagonal dominant node to the grounded node. It is unnecessary for each virtual capacitor to be inserted to the node having diagonal dominant to avoid redundant process as inserting the capacitor. In other words, the value of the adaptive virtual capacitors for the diagonal dominant node is zero. The linearized nodal equations are mapped by inserting adaptive virtual capacitors to the state equation as

$$\mathbf{C}_{virtual}\frac{d\tilde{\mathbf{v}}_k^{m+1}}{dt_{virtual}} = -\mathbf{J}\cdot\tilde{\mathbf{v}}_k^{m+1} + \mathbf{f}_k \tag{5}$$

where $\mathbf{C}_{virtual}$ is decided as making $\mathbf{J}$ more diagonal dominant. The solution of the linearized equation (4) is derived as a steady solution of the virtual state equation (5). The virtual state equation should be numerically integrated by the BEM. Each virtual capacitor for the diagonal virtual matrix $\mathbf{C}$ is inserted based on

$$S_i = \sum_{j=1, j\neq i}^{n} | J_{ij}| \tag{6}$$

where the S_i is sum of absolute value of non diagonal part of $\mathbf{J}$. Each capacitor C_i is determined by

$$C_i = \begin{cases} 0 & (J_{ii} \geq L \times S_i) \\ L \times S_i - J_{ii} & (J_{ii} < L \times S_i). \end{cases} \tag{7}$$

where J_{ij} is (i, j) elements of Jacobian matrix and L is a parameter that expresses the level of diagonal dominant. The value of the parameter is determined as $L = 1.25$ in this work. The C_i is virtual capacitor. The proposed relaxation method is configured by triple loops of NRM, BEM and GSM as shown in Fig.1. The SRP guarantee the parallelism in this work by keeping the structure of the Jacobian matrix.

3 Fast Automatic Differentiation

The main objectives of this paper is to model and analyze the neural networks having complex nonlinear and PWL functions. And the FAD is applied here to analyze such complex nonlinear functions. The performance of these calculations depends greatly on the accuracy of the partial derivatives that make up the Jacobian matrix and on the efficiency by which they are computed [7]. It can be done by going through the computational graph of Fig.2. From the viewpoint

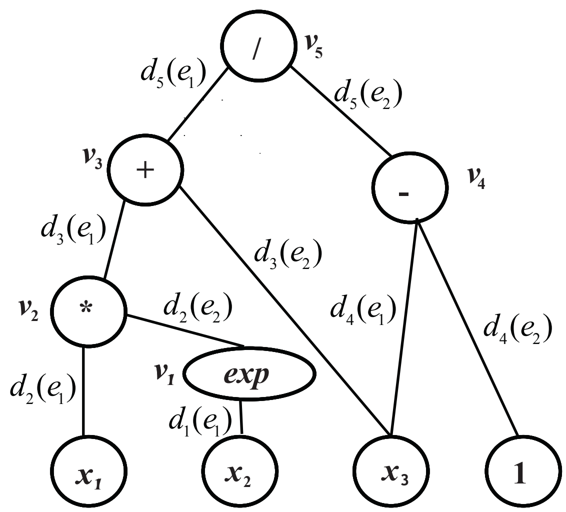

Figure 2: The computational graph.

of graph, the computation of partial derivative function by FAD is a shortest route through a non-rotating graph.

$$\mathbf{f}(x_1, x_2, x_3) = \frac{x_1 * exp(x_2) + x_3}{x_3 - 1}. \qquad (8)$$

Let's suppose the program has been assigned that halts at the finite time to compute the function similar to equation (8). These partial derivative obtained from the graph theory makes the Jocobian Matrix which is later on applied to NRM of the SRP.

4 Simulations and Results

4.1 LC Independent Oscillating Neural Networks

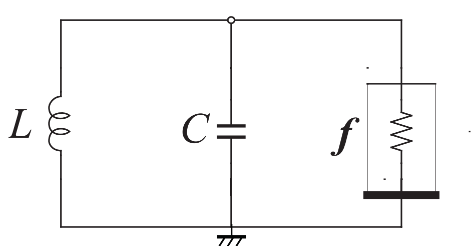

Figure 3: A LC model neuron with the function f.

To realize the LC independent oscillating neural network with the functions $f = (-v^3 + v)/sin(v)$ and $f = (v^3 - v)/cos(v)$, we simulated the neuron as in the Fig.3. The simulation result is shown in the Fig.4 analyzed by R-FAD. The same simulation was conducted with Runge-Kutta method and the

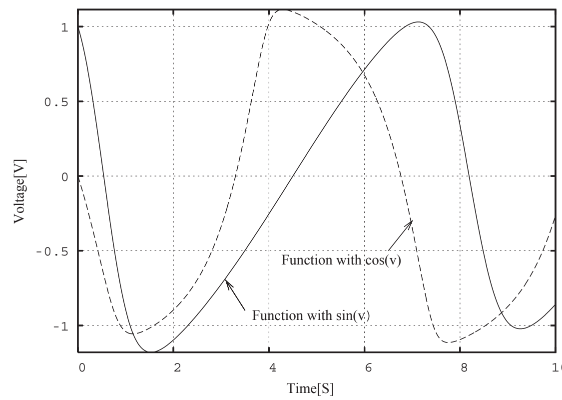

Figure 4: The R-FAD simulation result of LC oscillating neurons for functions $f = (-v^3 + v)/sin(v)$ and $f = (v^3 - v)/cos(v)$ with L=1[H], C=1[F].

result was found same as R-FAD. By the simulation result of Fig.4, it can be said R-FAD is found appropriate for the modeling and analysis of the neural networks having complex nonlinear function in it. The oscillating neurons with the much complex functions $f = (0.001v^9 - 0.5v)/(1.2 - sin(v \times 1.1))$, $f = (0.001v^9 - 0.5v)/(1.2 - cos(v \times 0.5))$ and $f = (0.001v^9 - 0.5v)$ was simulated. The simulation result is shown in Fig.5. We can see clearly the nonlinear wave propagation.

4.2 Oscillating CNN and Synchronization

We have carried out the simulation for the oscillation of a CNN by applying complex nonlinear functions by R-FAD with the L and C in the macro model form. The value of L and C are $200\mu[H]$ and $20\mu[F]$ respectively. The main objectives of this simulation are to see the synchronization of the neurons with small modulations by $sin(\cdot)$ and $cos(\cdot)$.

The Fig. 6 shows neurons placed with odd numbers as 1,3,5,7,9,11,13,15 having a complex nonlinear function $f = (0.0001v^3 + 0.09v)/(1.7 - sin(v \times 2))$ as well as the neurons placed with even numbers as 2,4,6,8,10,12,14,16 having a complex nonlinear function $f = (0.0001v^3 + 0.09v)/(1.7 - cos(v \times 2))$. Here, at the random initial values, the phase transient phenomenon has taken place at first. However, while the neurons keep on oscillating they overlap with each other toward the direction of synchronization.That is, a new discovery here is that there are clear synchronizing phenomena for direction of amplitudes after transient time region. Thus, the R-FAD can be used for the analysis of nonlinear neural synchronization which is important from the biological and environmental viewpoints. The same simulation was con-

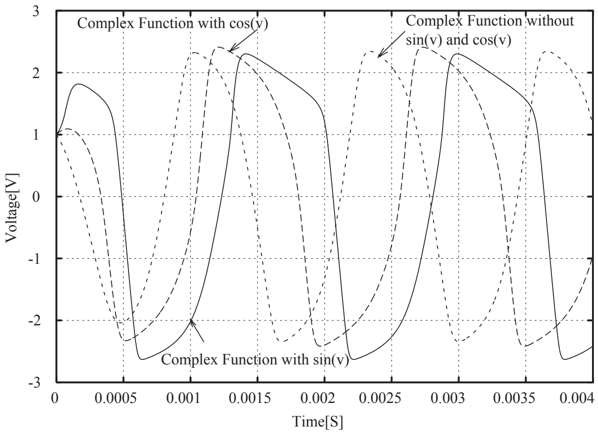

Figure 5: The R-FAD simulation result of LC Oscillating neurons having complex nonlinear functions $f = (0.001v^9 - 0.5v)/(1.2 - sin(v \times 1.1))$, $f = (0.001v^9 - 0.5v)/(1.2 - cos(v \times 0.5))$ and $f = (0.001v^9 - 0.5v)$ with $L = 2\mu[H]$, $C = 200\mu[F]$.

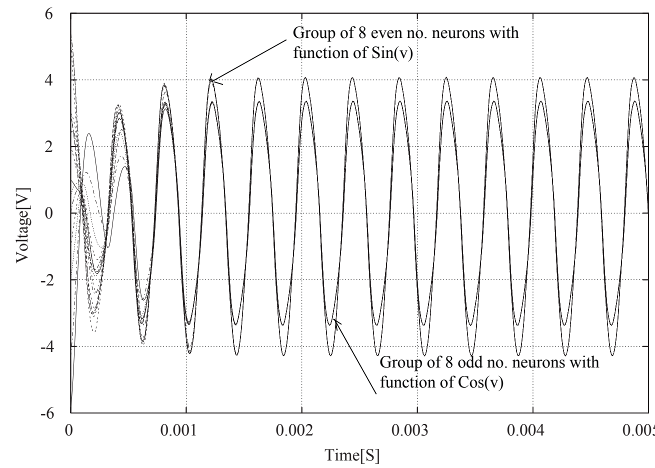

Figure 6: The synchronization of the neurons having random initial value. The functions are $f = (0.0001v^3 + 0.09v)/(1.7 - sin(v \times 2))$ and $f = (0.0001v^3 + 0.09v)/(1.7 - cos(v \times 2))$.

ducted with Runge-Kutta method and the result was found closely same as R-FAD.

5 Conclusion

The proposed SRP is expected to be used generally as a new R-FAD which models and analyzes large circuit with neurons having complex nonlinear and PWL functions. The proposed R-FAD has parallelism by relaxation-based algorithm keeping the structure of Jacobian matrix and general expression of nonlinearity. That is, the FAD can compute complex nonlinear functions by constructing the Jacobian matrix with its elements automatically. The important points in the proposed R-FAD are that the FAD, which is not used in the SPICE, can be used and that the connection of neural circuits which must be defined by differential equations can be defined explicitly in the input netlist file. We discovered interesting synchronization. As our future work, we will develop the R-FAD that can analyze the more multi-dimensional current variable i as $f(v_1, \cdots, v_l)$. And, also we will add learning algorithm. The executable file is opened on web (http://www.tlab.ee.sophia.ac.jp/SDP).

Acknowledgements: This research is suported by the fund of Open Research Center Project from MEXT of Japanese Government (2002-2006).

References:

[1] L.W. Nagal, "SPICE2; A computer program to simulate semiconductor circuits, *Electronics Research Laboratory*, memorandum no. ERL-M520, The University of California, Berkeley, 9 May. 1975.

[2] A. Ushida and M.Tanaka, Computer Simulations of Electronics Circuits, *Corona Publishing Co.*, Tokyo Japan 2002.

[3] Kubota and Iri, Automatic Differentiation of Algorithms and Applications, *Corona Publishing Co.*, Tokyo Japan.

[4] M. Iri and K. Kubota, Norms, rounding errors, partial derivatives and fast automatic differentiation, *IEICE Trans.* vol.E74, no.3, pp.463-471, March.1991.

[5] M.Asai and T. Mamoru, Relaxation-based algorithm for analysis of large nonlinear networks, *IEICE Trans.* vol.J70-A no.1, pp.82-92 Jan. 1987".

[6] M. Tanaka and M. Asai, Relation-based algorithm by variable virtual capacitors for large stiff nonlinear networks, *Proc. ISACA'87*, pp.617-620, 1987.

[7] J. E. Tolsma and P. L. Barton, Efficient calculation of sparse jacobians, *Siam J. Sci. Comput*, vol.20, no.6 pp.2282-2296, 1999.