New Multilevel Newton-Raphson Method for Parallel Circuit Simulation

Mikko Honkala*, Janne Roos*, and Martti Valtonen*

Abstract — **In this paper, a new variant of the multilevel Newton–Raphson method for parallel circuit simulation is presented. Its local and global convergence properties are studied. It is shown how, with specific circuit equation formulation, the multilevel method can be adjusted in order to achieve better global convergence. Finally, experimental results are presented.**

1 Introduction

In circuit simulators, Newton–Raphson (NR) iteration is a widely used method for solving nonlinear circuit equations. The parallel computation of this algorithm has roots in decomposed circuit analysis called diakoptics or tearing [1, 2], which can be performed by hierarchical LU-factorization [3]. But, especially in networked parallel computation, communication between processors is a bottleneck [4], and there are multilevel Newton analysis [5] type methods, better suitable for parallel processing [4, 6, 7].

In this paper, a new variant of the multilevel Newton–Raphson (MLNR) method for efficient parallel circuit simulation is presented. The main idea is to modify the iteration such that good global convergence properties can be achieved. The local quadratic convergence of the method is proved.

The new MLNR method has been implemented in the in-house developement version of APLAC circuit simulation and desing tool [8], and its performance has been verified with simulations.

2 Circuit equation formulation

In parallel circuit simulation, the circuit is partitioned into subcircuits and a main circuit consisting of the connections between subcircuits, which can be whatever circuit elements or connection nodes only. The circuit equations can be created, e.g., using nodal formulation or modified nodal analysis (MNA) [9]. In APLAC, the nonlinear MNA equations are reduced to the nonlinear nodal formulation using the gyrator transform [10].

The system of nonlinear nodal equations can be written as

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

where $\mathbf{x} \in \mathbb{R}^n$ are nodal voltages of the circuit and $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ has a Jacobian matrix $\mathbf{J}$. The nodal equations are, typically, solved using the NR method which is sequence of iterations

$$\mathbf{x}^{k+1} = \mathbf{x}^k - (\mathbf{J}^k)^{-1}\mathbf{f}(\mathbf{x}^k), \tag{2}$$

where k is the iteration index.

Consider a circuit which has n nodes and which can be decomposed into m subcircuits consisting of n_i internal nodes and $n_{\mathrm{E}i}$ external connection nodes. The nonlinear system of nodal equations for internal and external nodes can be written as

$$\begin{aligned} \mathbf{f}_i(\mathbf{x}_i, \mathbf{x}_{\mathrm{E}}) &= 0, \\ \mathbf{f}_{\mathrm{E}}(\mathbf{x}_1, \ldots, \mathbf{x}_m, \mathbf{x}_{\mathrm{E}}) &= 0, \end{aligned} \tag{3}$$

respectively, where $i = 1, \ldots, m$, and $\mathbf{x}_i \in \mathbb{R}^{n_i}$ are internal nodal voltages of subcircuits, $\mathbf{x}_{\mathrm{E}} \in \mathbb{R}^{n_{\mathrm{E}}}$ voltages of external connection nodes of subcircuits, $\mathbf{f}_i : \mathbb{R}^{n_i} \times \mathbb{R}^{n_{\mathrm{E}}} \to \mathbb{R}^{n_i}$, and $\mathbf{f}_{\mathrm{E}} : \mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_{\mathrm{E}}} \to \mathbb{R}^{n_{\mathrm{E}}}$.

The Jacobian matrix $\mathbf{J}$ has a bordered block diagonal (BBD) form [3]:

$$\mathbf{J} = \begin{bmatrix} \mathbf{A}_1 & & & & \mathbf{B}_1 \\ & \mathbf{A}_2 & & & \mathbf{B}_2 \\ & & \ddots & & \vdots \\ & & & \mathbf{A}_m & \mathbf{B}_m \\ \mathbf{C}_1 & \mathbf{C}_2 & \ldots & \mathbf{B}_m & \mathbf{D} \end{bmatrix}, \tag{4}$$

where

$$\begin{aligned} \mathbf{A}_i &= \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i} \in \mathbb{R}^{n_i \times n_i}, & \mathbf{B}_i &= \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_{\mathrm{E}}} \in \mathbb{R}^{n_i \times n_{\mathrm{E}}}, \\ \mathbf{C}_i &= \frac{\partial \mathbf{f}_{\mathrm{E}}}{\partial \mathbf{x}_i} \in \mathbb{R}^{n_{\mathrm{E}} \times n_i}, & \mathbf{D} &= \frac{\partial \mathbf{f}_{\mathrm{E}}}{\partial \mathbf{x}_{\mathrm{E}}} \in \mathbb{R}^{n_{\mathrm{E}} \times n_{\mathrm{E}}}. \end{aligned} \tag{5}$$

$\mathbf{f}_{\mathrm{E}}$, as well as $\mathbf{D}$, can be further decomposed into parts, that contain the contributions of the circuit elements of the main circuit and each subcircuit:

$$\mathbf{f}_{\mathrm{E}} = \mathbf{f}_{\mathrm{E}0} + \sum_{i=1}^{m} \mathbf{f}_{\mathrm{E}i}, \tag{6}$$

$$\mathbf{D} = \mathbf{D}_{\mathrm{E}0} + \sum_{i=1}^{m} \mathbf{D}_i. \tag{7}$$

*Helsinki University of Technology, Circuit Theory Laboratory, P.O.Box 3000, FIN-02015 HUT, Finland. Tel: +358-9-4512297, Fax: +358-9-4514818. Email: {mahonkal,janne,martti}@aplac.hut.fi

The conventional NR iteration (2) can be performed in parallel by solving Eqs. (3)–(7) using hierarchical LU-factorization [3].

In the following, these equations will be modified such that we take the short-circuit currents flowing from the main circuit to the subcircuits as extra variables (see Fig. 1).

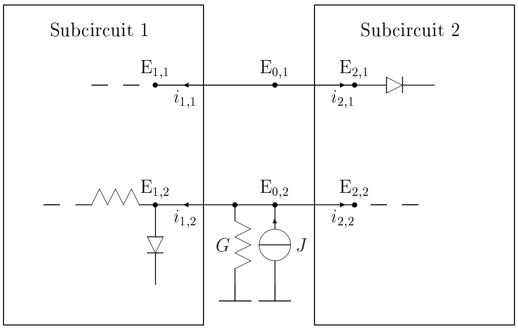

Figure 1: Currents flowing from main circuit to subcircuits.

The idea of this formulation is that, this way, each subcircuit has independent short-circuit currents $\mathbf{i}_i \in \mathbb{R}^{n_{\mathrm{E}i}}$ and external variables $\mathbf{x}_{\mathrm{E}i} \in \mathbb{R}^{n_{\mathrm{E}i}}$, and the only connection is at the main circuit level. Thus the subcircuits do not have common variables. The nodal equations for internal, subcircuit connection nodes and main circuit nodes take the form

$$\begin{aligned} \mathbf{f}_i(\mathbf{x}_i, \mathbf{x}_{\mathrm{E}i}) &= 0, \\ \mathbf{f}_{\mathrm{E}i}(\mathbf{x}_i, \mathbf{x}_{\mathrm{E}i}) + \mathbf{i}_i &= 0, \\ \mathbf{f}_{\mathrm{E}0}(\mathbf{x}_{\mathrm{E}0}, \mathbf{x}_{\mathrm{E}_\mathrm{I}}, \mathbf{i}_\mathrm{I}) &= 0, \end{aligned} \tag{8}$$

respectively, where $i = 1, \ldots, m$, subscript I denotes all i, $\mathbf{f}_{\mathrm{E}i} : \mathbb{R}^{n_i} \times \mathbb{R}^{n_{\mathrm{E}i}} \to \mathbb{R}^{n_{\mathrm{E}i}}$, and $\mathbf{f}_{\mathrm{E}0} : \mathbb{R}^{n_{\mathrm{E}0}} \times \mathbb{R}^{n_{\mathrm{E}_\mathrm{I}}} \times \mathbb{R}^{n_{\mathrm{E}_\mathrm{I}}} \to \mathbb{R}^{n_{\mathrm{E}0}}$.

The advantage of this formulation is presented in Section 4.2, where global convergence conditions are derived for the new MLNR method with short-circuit currents.

3 New Multilevel Newton–Raphson method

The new MLNR method is modified from the multilevel Newton analysis method presented in Ref. [5]. In the method in Ref. [5], instead of taking global NR steps as in Eq. (2), the iterations are taken at multiple levels. Between outer iterations, the external variables are kept constant and only inner variables of subcircuits are iterated:

$$\mathbf{x}_i^{k,j+1} = \mathbf{x}_i^{k,j} - \left(\mathbf{A}_i^{k,j}\right)^{-1} \mathbf{f}_i(\mathbf{x}_i^{k,j}, \mathbf{x}_{\mathrm{E}i}^{k}), \tag{9}$$

where j is the inner iteration index. The inner iteration is stopped at some error level and then the main-circuit variables are iterated using subcircuits as macromodels. The original method is not the most suitable for efficient parallel processing, because the numbers of inner iterations are not balanced and the global convergence cannot be easily adjusted. MLNR type methods which are more suitable for parallel processing have been presented in [4, 6, 7]. The benefits of these multilevel methods are lower communication between processors and fast upper-level convergence.

In the new MLNR method proposed, the main new idea is to modify the iteration such that the global convergence can be easily controlled while the local convergence of all variables is quadratic. The load balancing is performed by taking only J inner iterations between each outer NR step (as it is done also in Ref [7]). By taking global iteration steps at the outer iteration level, i.e., by iterating all variables from $\mathbf{x}^{k,J}$ to $\mathbf{x}^{k+1,0}$ instead of updating only external variables, local quadratic convergence of the new method with only J inner iterations is achieved (see Section 4.1). Thus the iteration is speeded up. If the short-circuit currents are added (which is not necessary for the new MLNR method), we can adjust the inner iterations such that the the whole iteration has good global convergence (see Section 4.2).

Computation of the global NR step is more expensive than the outer iteration step where only external variables are updated, but this is compensated by faster convergence of the total iteration.

The new MLNR method (without $\boldsymbol{\Delta}\mathbf{x}$ adjusting) can be summarized as follows (possibly added short-circuit currents are imbedded into $\mathbf{f}$ and $\mathbf{J}$):

Algorithm 1.

1. Set $\mathbf{x}^0$, ε and J.
2. Begin outer iteration: Set $k = 0$.
    - (a) Begin inner iterations for subsystems i: Set $j = 0$.
    - (b) Solve $\mathbf{A}_i^{k,j} \boldsymbol{\Delta}\mathbf{x}_i^{k,j} = -\mathbf{f}_i(\mathbf{x}_i^{k,j}, \mathbf{x}_{\mathrm{E}i}^{k})$.
    - (c) Set $\mathbf{x}_i^{k,j+1} = \mathbf{x}_i^{k,j} + \boldsymbol{\Delta}\mathbf{x}_i^{k,j}$.
    - (d) End inner iterations after J iterations.
3. Take global iteration step: $\mathbf{x}^{k+1,0} = \mathbf{x}^{k,J} - (\mathbf{J}^{k,J})^{-1}\mathbf{f}(\mathbf{x}^{k,J})$.
4. End outer iteration if $\|\Delta\mathbf{x}\| < \varepsilon$.

The new MLNR iteration with zero inner iterations reduces to normal NR iteration.

4 Convergence

4.1 Local Convergence

Let $\mathbf{f}$ be differentiable in an open set $\Omega \subset \mathbb{R}^n$ and let $\mathbf{J} : \Omega \to \mathbb{R}^{n \times n}$. $\mathbf{J}$ is Lipschitz continous. We also assume that a solution $\mathbf{x}^* \in \Omega$ exists and that $\mathbf{J}(\mathbf{x}^*)$ is nonsingular. Under these assumptations, if we start the NR iteration "sufficiently close" to the solution (on Ω), the convergence is quadratic,

$$\|\mathbf{e}^{k+1}\| \leq K\|\mathbf{e}^k\|^2, \tag{10}$$

where K is a positive constant and $\mathbf{e} = \mathbf{x} - \mathbf{x}^*$. The proof can be found, e.g., in Ref. [11].

In the following, some convergence properties of the method proposed will be shown. Therefore, we assume that there is an L such that $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}^*)\| \leq L\|\mathbf{x} - \mathbf{x}^*\|$, and that for some M $\|(\mathbf{A}^{k,j})^{-1}\| \leq M$. For the convergence theorem, the following lemma is needed.

Lemma 1: It follows from the assumptions that

$$\|\mathbf{e}^{k,j}\| \leq \hat{K}\|\mathbf{e}^{k,0}\|, \tag{11}$$

where $\hat{K} = (1 + LM)^j$, for all $j = 1, 2, \ldots$

Proof: (By induction.) $j = 1$: The inner iteration (9) can be rewritten, by adding $\mathbf{e}^k_{\mathrm{E}}$ and $\mathbf{f}^k_{\mathrm{E}}$, as follows:

$$\begin{bmatrix} \mathbf{e}^{k,1}_{\mathrm{I}} \\ \mathbf{e}^k_{\mathrm{E}} \end{bmatrix} = \begin{bmatrix} \mathbf{e}^{k,0}_{\mathrm{I}} \\ \mathbf{e}^k_{\mathrm{E}} \end{bmatrix} - \begin{bmatrix} (\mathbf{A}^{k,0})^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{f}^{k,0}_{\mathrm{I}} \\ \mathbf{f}^k_{\mathrm{E}} \end{bmatrix}. \tag{12}$$

Then

$$\begin{aligned} \left\| \begin{bmatrix} \mathbf{e}^{k,1}_{\mathrm{I}} \\ \mathbf{e}^k_{\mathrm{E}} \end{bmatrix} \right\| &= \left\| \begin{bmatrix} \mathbf{e}^{k,0}_{\mathrm{I}} \\ \mathbf{e}^k_{\mathrm{E}} \end{bmatrix} - \begin{bmatrix} (\mathbf{A}^{k,0})^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{f}^{k,0}_{\mathrm{I}} \\ \mathbf{f}^k_{\mathrm{E}} \end{bmatrix} \right\| \\ &\leq \left\|\mathbf{e}^{k,0}\right\| + \left\|(\mathbf{A}^{k,0})^{-1}\right\| \left\|\mathbf{f}^{k,0}\right\| \\ &\leq (1 + LM) \left\|\mathbf{e}^{k,0}\right\|, \end{aligned} \tag{13}$$

and thus Eq. (11) is true for $j = 1$. The inductive hypothesis is that

$$\left\|\mathbf{e}^{k,j}\right\| \leq (1 + LM)^j \left\|\mathbf{e}^{k,0}\right\|. \tag{14}$$

Using the hypothesis for $j + 1$,

$$\begin{aligned} \left\|\mathbf{e}^{k,j+1}\right\| &\leq \left\|\mathbf{e}^{k,j}\right\| + \left\|(\mathbf{A}^{k,j})^{-1}\right\| \left\|\mathbf{f}^{k,j}\right\| \\ &\leq \left\|\mathbf{e}^{k,j}\right\| + LM \left\|\mathbf{e}^{k,j}\right\| \\ &\leq (1 + LM)^{j+1} \left\|\mathbf{e}^{k,0}\right\|. \end{aligned} \tag{15}$$

Lemma 1 holds for all j by induction. □

If we start the iteration inside the ball defined by

$$B(\delta) = \{\mathbf{x} \mid \|\mathbf{e}\| < \delta\}, \tag{16}$$

the following theorem provides local quadratic convergence of the new MLNR method.

Theorem 1: If the assumptions hold, there is $\delta > 0$ such that if $\mathbf{x}^{0,0} \in B(\delta)$, then the new MLNR method converges quadratically.

Proof: Let δ be small enough so that $B(\hat{K}\delta) \subset \Omega$. Reduce, if needed, δ such that $K\hat{K}^2\delta = \eta < 1$. If $\mathbf{x}^{k,0} \in B(\delta)$, then Eq. (10) and Lemma 1 imply that after outer iteration,

$$\begin{aligned} \|\mathbf{e}^{k+1,0}\| &\leq K\|\mathbf{e}^{k,J}\|^2 \leq K\hat{K}^2\|\mathbf{e}^{k,0}\|^2 \\ &\leq K\hat{K}^2\delta\|\mathbf{e}^{k,0}\| \leq \eta\|\mathbf{e}^{k,0}\| \\ &< \|\mathbf{e}^{k,0}\|, \end{aligned} \tag{17}$$

and $\mathbf{x}^{k+1,0} \in B(\eta\delta) \subset B(\delta)$. Since $\mathbf{x}^{0,0} \in B(\delta)$, $\mathbf{x}^k$ converges to $\mathbf{x}^*$ quadratically. □

4.2 Global Convergence

Eventhough the local convergence of the new method is quadratic, the global convergence is not guaranteed. In particular, finding the DC solution of a circuit leads easily to divergence due to a poor initial guess. In order to obtain global convergence, the step size of every outer iteration step (step 3 of Algorithm 1) can be adjusted such that at least $\|\mathbf{f}(\mathbf{x}^{k+1,0})\| < \|\mathbf{f}(\mathbf{x}^{k,J})\|$. It is even better if every inner iteration reduces the error, i.e., if $\|\mathbf{f}(\mathbf{x}^{k,j+1})\| < \|\mathbf{f}(\mathbf{x}^{k,j})\|$.

For the normal NR iteration, a large number of heuristic step-size damping methods [12] , which are usually based on *a priori* knowledge of nonlinear characteristics, have been presented. Also norm-reduction methods [12], which are 1D line search methods in the direction of the NR update, are efficiently used with NR iteration.

In the following, it will be shown how the convergence of the new MLNR method can be controlled by using step-size adjusting methods and short-circuit currents, such that every inner and outer iteration reduces the total error.

Because of short-circuit currents, all $\mathbf{f}_i$ and $\mathbf{f}_{\mathrm{E}i}$ are independent of variables of other subcircuits and $\mathbf{f}_{\mathrm{E}_0}$ depends on only $\mathbf{x}_{\mathrm{E}_0}$, $\mathbf{x}_{\mathrm{E}_\mathrm{I}}$ and $\mathbf{i}_\mathrm{I}$ which do not change during the inner iteration. Therefore, $\mathbf{f}_{\mathrm{E}_0}$ is constant during the inner iterations and we can investigate whether the norm of the left hand side of the nodal equations (8) decreases by checking whether the condition

$$\left\| \begin{bmatrix} \mathbf{f}^{k,j+1}_i \\ \mathbf{f}^{k,j+1}_{\mathrm{E}i} + \mathbf{i}^k_i \end{bmatrix} \right\| < \left\| \begin{bmatrix} \mathbf{f}^{k,j}_i \\ \mathbf{f}^{k,j}_{\mathrm{E}i} + \mathbf{i}^k_i \end{bmatrix} \right\| \tag{18}$$

is satisfied. If the condition is not satisfied, the inner iteration should be stopped at the point where the error was decreased. Damping methods can be used for each inner iteration to achieve global convergence. In practice, line-search methods can be used within the inner iteration, but there is no theoretical justification for this because the direction of the inner iteration step is not necessarily in the global descent direction.

After J iterations, the outer-iteration step starts from the point where the error is smaller than or equal to the starting point of the inner iteration, and because, then, the outer iteration step is a normal NR step, it is in the direction of steepest descent and both damping and norm-reduction methods can be used for step-size adjusting.

5 Simulation results

The new MLNR method with convergence aiding methods has been implemented in the in-house developement version of APLAC [8], where sparse matrix reordering and factorization routines are used for Jacobian matrices. The parallel processing routines have been implemented using the Parallel Virtual Machine (PVM) package [13].

For convergence aiding, the maximum step-size is set to 1 V. The diode damping method [14], which calculates the damping due the exponential current characteristics of diodes, and norm reduction method [14] for the outer iteration are used.

The example simulations have been done using 3 PCs in a local area network. The preprosessing phase includes input-file intepretation and symbolic reordering of sparse matrices.

The example circuit, which has 1440 bipolar transistors and 7746 nodes, is decomposed into 3 subcircuits. The number of outer iterations K versus number of inner iterations J and the simulation times are presented in Table 1. The unparallelized simulation with 19 iterations took 17.9 seconds.

J	K	preproc.	iter.	total
0	19	9.0 s	2.6 s	11.6 s
1	13	9.0 s	2.6 s	11.6 s
2	6	9.0 s	1.9 s	10.9 s
3	5	9.0 s	1.7 s	10.7 s

Table 1: Number of outer iterations K and simulation times versus number of inner iterations J.

6 Conclusion

The new multilevel Newton–Raphson method for parallel circuit simulation has been presented. Its good convergence properties have been demonstrated and verified with simulations. The reduction of iteration times may be more significant in other analyses, say, harmonic balance, where the computational load of iterations dominates the preprocessing.

References

[1] H. H. Happ, *Diakoptics and Networks*, Academic Press, New York, 1971.

[2] F. F. Wu, “Solution of Large-Scale Networks by tearing”, *IEEE Trans. Circuits Syst.*, vol. CAS-23, no. 12, pp. 706–713, Dec, 1976.

[3] M. Vlach, “LU Decomposition Algorithms for Parallel and Vector Computation” in *Analog Methods for Computer-Aided Circuit Analysis and Design*, (T. Ozawa, ed.), pp. 37–64, Marcel Deccer inc, 1988.

[4] N. Fröhlich, B. M. Riess, U. A. Wever and Q. Zheng, “A New Approach for Parallel Simulation of VLSI Circuits on a Transistor Level”, *IEEE Trans. Circuits Syst. I*, vol 45, no. 6, pp. 601–613, june 1998.

[5] N. B. G. Rabbat, A. L. Sangiovanni-Vincentelli and H. Y. Hsieh, “A Multilevel Newton Algorithm with Macromodeling and Latency for the Analysis of Large-Scale Nonlinear Circuits in the Time Domain”, *IEEE Trans. Circuits Syst.* vol. CAS-26, no. 9, pp. 733–741, Sep 1979.

[6] W. L. Engl, R. Laur, and H. K. Dirks, “MEDUSA — A Simulator for Modular Circuits”, *IEEE Tran. Computer-Aided Design*, vol. CAD-1, no. 2, pp. 85–93, Apr. 1982.

[7] X. Zhang, R. H. Byrd and R. B. Schnabel, “Parallel Methods for Solving Nonlinear Block Bordered System of Equations”, *SIAM j. Sci. Stat. Comput.*, vol. 13, no. 4, pp. 841–859, July 1992.

[8] `http://www.aplac.hut.fi/aplac/`

[9] C.-W. Ho, A. E. Ruehli, and P. A. Brennan, “The Modified Nodal Approach to Network Analysis”, *IEEE Trans. Circuits Syst.*, vol. 22, no. 6, pp. 504–509. June 1975.

[10] H. Gaunholt, P. Heikkilä, K. Mannersalo, V. Porra, and M. Valtonen, “Gyrator Transformation — A Better Way for Modified Nodal Approach”, *Proceedings of ECCTD’91*, Copenhagen, July 1991, vol. 2, pp. 864–872.

[11] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.

[12] J. Ogrodzki, *Circuit Simulation Methods and Algorithms*, CRC Press, USA, 1994.

[13] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek and V. Sunderam, *PVM: Parallel Virtual Machine, A Users’ Guide and Tutorial for Networked Parallel Computing*, The MIT Press, 1994.

[14] M. Valtonen, P. Heikkilä, H. Jokinen, and T. Veijola, “APLAC – object-oriented circuit simulator and design tool”, *Low-power HF microelectronics: a unified approach*, (G. A. S. Machado, ed.), pp. 333–372 IEE, London, 1996.