Optimization of Parallel Multilevel-Newton Algorithms on Workstation Clusters

Robert Gräb¹, Michael Günther¹ *, Utz Wever² and Qinghua Zheng²

¹ Technische Hochschule Darmstadt, Fachbereich Mathematik, Schloßgartenstr. 7, D-64289 Darmstadt, Germany
² Siemens AG, Corporate Research and Development, D-81730 München, Germany

Abstract. The design of dynamic memories and digital circuits demands the numerical simulation of networks with more than 10 000 transistors. Only a parallel implementation of industrial simulation packages allows short design cycles required today. Multilevel-Newton algorithms (MLN) based on circuit partitioning techniques have proved to be very efficient in parallel circuit simulation on workstation clusters. In industrial environments, the performance of these algorithms depends essentially on an optimal partitioning and adaptation to actually achieved transmission performance of the network. Recent results with the circuit simulator TITAN at Siemens' research laboratories are discussed.

1 Introduction

Circuit simulation programs are standard tools for computer aided design of electric circuits. At Siemens AG a major application is the verification of dynamic memories and digital circuits with more than 10 000 transistors. Due to the desired accuracy and reliability, classical circuit simulation algorithms such as solutions of nonlinear equations, numerical integration and exploitation of latency are used. They are implemented in the circuit simulator TITAN (SPICE [5] compatible input language), which was developed at Siemens' research laboratories, see [1].

In the past supercomputers have been used for simulating circuits. Nowadays the floating point performance of modern workstations approaches those of small vector computers. Thus it makes sense to utilize the overall power of a workstation cluster by running additional parallel software, see [4, 7]. But as the network performance is very low, only those algorithms guaranteeing a minimum of communication overhead can reach good speedup results. Domain decomposition methods split up the problem in the physical domain. Each domain is calculated by a separate process and can thus run in parallel. The boundaries of the domains are equalized by a serial master process. In [7, 8], an improvement of multilevel Newton methods was developed, which takes into account the low communication of workstation clusters. A further decrease of communication by

* Corresponding author. E-Mail: guenther@mathematik.th-darmstadt.de
introducing Quasi-Newton techniques was considered in [8]. In this paper we aim at an additional optimization with respect to parallel computation costs.

In section 2, multilevel Newton methods for circuit simulation are introduced. They exploit iteration latency and can be optimized in three directions: in case of low network performance, alternative tangential corrections can be used, see section 3. The optimal partitioning of an electric circuit is discussed in section 4. Additionally in section 5, the number of inner iterations is adapted to the effective transfer rate of data.

2 Parallel Multilevel-Newton Algorithms

By applying the scheme of Modified Nodal Analysis to a given circuit, a system of \( N \) nonlinear differential algebraic equations is setup:

\[
\begin{align*}
\mathbf{f}(\mathbf{q}(\mathbf{x}(t)), \mathbf{x}(t), t) &= \left( \begin{array}{c}
f_1(\mathbf{x}(t), t) + \mathbf{q}(\mathbf{x}(t)) \\
f_2(\mathbf{x}(t), t)
\end{array} \right) = 0.
\end{align*}
\]

\( f_1 \) and \( f_2 \) describe the static part of the network equations, \( q \) the charges and magnetic fluxes and \( \mathbf{x} \in \mathbb{R}^N \) the node potentials and the branch currents through voltage sources and inductors. An implicit time discretization scheme such as the Backward Differentiation Formula yields to nonlinear equations, see [1]:

\[
\mathbf{F}(\mathbf{x}) = 0. \tag{1}
\]

A standard method to handle this equation is Newton’s method, where the linear system is solved by Gauss’ algorithm for sparse matrices. By applying domain decomposition methods, equation (1) can be decoupled into the nonlinear system

\[
\begin{align*}
F_i(\mathbf{x}_i, \mathbf{x}_{m+1}) &= 0 \quad \text{for} \quad i = 1, \ldots, m, \quad (2) \\
G(\mathbf{x}_1, \ldots, \mathbf{x}_m, \mathbf{x}_{m+1}) &= 0 \quad (3)
\end{align*}
\]

with \( F_i \in \mathbb{R}^{N_i} \), \( \mathbf{x}_i \in \mathbb{R}^{N_i} \), \( \mathbf{x}_{m+1} \in \mathbb{R}^{N_{m+1}} \) and \( G \in \mathbb{R}^{N_{m+1}} \). \( F_i \) describes the \( i \)-th sub-circuit, depending only on the inner variables \( \mathbf{x}_i \) and coupling quantities \( \mathbf{x}_{m+1} \). The subsystems are coupled by the system \( G \) of generally small dimension. An efficient multilevel Newton algorithm for solving (2,3) reads, see [8]:

Multilevel-Newton algorithm — MLN(n)

\[
\begin{align*}
&\text{start: } (\mathbf{x}_1^0, \ldots, \mathbf{x}_m^0, \mathbf{x}_{m+1}^0); \ \text{latency} = \text{false} \\
&\text{do } l = 0, 1, 2, 3, \ldots \\
&\quad \text{do parallel } j = 1, \ldots, m \\
&\quad \quad \mathbf{x}_{j}^{l+1} = \mathbf{x}_{j}^{l} \\
&\quad \quad \text{do } i = 0, 1, 2, \ldots, n - 1 \\
&\quad \quad \quad \frac{\partial \mathbf{F}_j}{\partial \mathbf{x}_j} \Delta \mathbf{x}_j + \mathbf{F}_j(\mathbf{x}_j^{l+1}, \mathbf{x}_{m+1}^{l+1}) = 0; \ \mathbf{x}_j^{l+1} = \mathbf{x}_j^{l} + \Delta \mathbf{x}_j \\
&\quad \quad \text{end do}\_i \\
&\quad \quad \text{if ( latency = false ) then } S_j^l = \frac{\partial \mathbf{G}}{\partial \mathbf{x}_j} (\frac{\partial \mathbf{F}_j}{\partial \mathbf{x}_j})^{-1} \frac{\partial \mathbf{F}_j}{\partial \mathbf{x}_{m+1}} \\
&\quad \text{end do}\_j \\
&\quad \text{if ( latency = false ) then } S^l = \frac{\partial \mathbf{G}}{\partial \mathbf{x}_{m+1}} - \sum_{j=1}^{m} S_j^l = L^l U^l
\end{align*}
\]