
Terminal Current Interpolation for Multirate Time Integration of Hierarchical IC Models

A. Verhoeven¹, E.J.W. ter Maten^{1,2}, J.J. Dohmen², B. Tasić²,
and R.M.M. Mattheij¹

¹ Eindhoven University of Technology (CASA), Den Dolech 2, 5612 AZ Eindhoven,
The Netherlands, Arie.Verhoeven@na-net.ornl.gov, www.casa.tue.nl

² NXP Semiconductors (Design Methods), Eindhoven, The Netherlands
www.nxp.com

1 Introduction

Multirate time-integration methods [3–5] appear to be attractive for initial value problems for DAEs with latency or multirate behaviour. Latency means that parts of the circuit are constant or slowly time-varying during a certain time interval, while multirate behaviour means that some variables are slowly time-varying compared to other variables. In both cases, it would be attractive to integrate these slow parts with a larger timestep than the other parts. This saves the computational workload while the accuracy is preserved. A nice property of multirate is that it does not use any linear structure, in contrast to MOR, but only a relaxation concept. If the coupling is sufficiently monitored and the partitioning is well chosen, multirate can be very efficient.

In this paper we will show how multirate time integration can be applied to hierarchical circuit models. Besides the classical interpolation variants, also some new implicit variants are discussed.

2 Hierarchical Circuit Models

Integrated Circuits can be modeled by a hierarchical system of differential-algebraic equations [1, 2]:

$$\frac{d}{dt}[\mathbf{q}(t, \mathbf{x})] + \mathbf{j}(t, \mathbf{x}) = \sum_{i=1}^N \mathbf{B}_{(i)}^T \left[\frac{d}{dt}[\mathbf{q}^{(i)}(t, \mathbf{x}^{(i)})] + \mathbf{j}^{(i)}(t, \mathbf{x}^{(i)}) \right] = \mathbf{0}. \quad (1)$$

Clearly this circuit model with global state vector $\mathbf{x} \in \mathbb{R}^d$ consists of N coupled subcircuit models. Each local state vector $\mathbf{x}^{(i)} \in \mathbb{R}^{d_i}$ (voltages, currents) consists of a terminal ($\hat{\mathbf{x}}^{(i)}$) and an internal ($\tilde{\mathbf{x}}^{(i)}$) part:

$$\mathbf{x}^{(i)} = \mathbf{B}_{(i)} \mathbf{x} = \begin{bmatrix} \hat{\mathbf{x}}^{(i)} \\ \check{\mathbf{x}}^{(i)} \end{bmatrix}.$$

The matrices $\mathbf{B}_{(i)} \in \{0, 1\}^{d_i \times d}$, defined by

$$\mathbf{B}_{(i)} = \begin{bmatrix} \hat{\mathbf{B}}_{(i)} & \\ & \check{\mathbf{B}}_{(i)} \end{bmatrix}, \quad (2)$$

are used to select the proper parts $\hat{\mathbf{x}}^{(i)} = [\hat{\mathbf{B}}_{(i)}] \mathbf{x}$ and $\check{\mathbf{x}}^{(i)} = [\check{\mathbf{B}}_{(i)}] \mathbf{x}$ of $\mathbf{x}^{(i)}$ from \mathbf{x} . This even allows for a hierarchical structure.

Similarly, also the functions $\mathbf{q}^{(i)}$ (charges, fluxes) and $\mathbf{j}^{(i)}$ (currents, voltages) have a similar structure: $\mathbf{q}^{(i)} = \mathbf{B}_{(i)} \mathbf{q} = \begin{bmatrix} \hat{\mathbf{q}}^{(i)} \\ \check{\mathbf{q}}^{(i)} \end{bmatrix}$, $\mathbf{j}^{(i)} = \mathbf{B}_{(i)} \mathbf{j} = \begin{bmatrix} \hat{\mathbf{j}}^{(i)} \\ \check{\mathbf{j}}^{(i)} \end{bmatrix}$.

We can rewrite (1) in a part consisting of the collected equations for the terminal unknowns (3) and a part consisting of the remaining equations for the internal unknowns (4):

$$\sum_{i=1}^N \hat{\mathbf{B}}_{(i)}^T \left[\frac{d}{dt} [\hat{\mathbf{q}}^{(i)}(t, \mathbf{x}^{(i)})] + \hat{\mathbf{j}}^{(i)}(t, \mathbf{x}^{(i)}) \right] = \mathbf{0}, \quad (3)$$

$$\frac{d}{dt} [\check{\mathbf{q}}^{(i)}(t, \mathbf{x}^{(i)})] + \check{\mathbf{j}}^{(i)}(t, \mathbf{x}^{(i)}) = \mathbf{0}, \quad i = 1, \dots, N. \quad (4)$$

Each subcircuit model can again be further decomposed in this manner.

3 Multirate Transient Analysis

For single-rate time integration all equations are discretised simultaneously by the same time step. If the time constants per subcircuits are quite different, it is attractive to perform multirate time integration. Then the fast subcircuits can be integrated on a local, fine, time-grid. Especially when the fast subcircuits are small in size, the additional costs for synchronisation and partitioning can be overcome and the overall multirate procedure becomes much more efficient than the single-rate time integration. An attractive multirate method is the Compound-Fast version [5, 6, 8], which first integrates the whole system at the new coarse time gridpoint and after that re-integrates only the active part at the fine time-grid. We will denote the coarse and fine time gridpoints by $\{T_n, 0 \leq n \leq N\}$ and $\{t_{n-1,m}, 1 \leq n \leq N, 0 \leq m \leq q_n\}$ with macro-steps $H_n := T_n - T_{n-1}$, and micro-steps $h_{n,m} := t_{n,m} - t_{n,m-1}$ and multirate factors q_n such that $t_{n-1,0} = T_{n-1}$, $t_{n-1,q_n} = T_n$. For a partitioning in a latent (slow) part, $\mathbf{x}^{(L)} \in \mathbb{R}^{d_L}$ and $\mathbf{x}^{(A)} \in \mathbb{R}^{d_A}$, we typically get:

$$\begin{aligned} \hat{\mathbf{B}}_{(L)}^T \left[\frac{d}{dt} [\hat{\mathbf{q}}^{(L)}(t, \mathbf{x}^{(L)})] + \hat{\mathbf{j}}^{(L)}(t, \mathbf{x}^{(L)}) \right] + \hat{\mathbf{B}}_{(A)}^T \left[\frac{d}{dt} [\hat{\mathbf{q}}^{(A)}(t, \mathbf{x}^{(A)})] + \hat{\mathbf{j}}^{(A)}(t, \mathbf{x}^{(A)}) \right] &= \mathbf{0}, \\ \frac{d}{dt} [\tilde{\mathbf{q}}^{(L)}(t, \mathbf{x}^{(L)})] + \check{\mathbf{j}}^{(L)}(t, \mathbf{x}^{(L)}) &= \mathbf{0}, \quad \frac{d}{dt} [\tilde{\mathbf{q}}^{(A)}(t, \mathbf{x}^{(A)})] + \check{\mathbf{j}}^{(A)}(t, \mathbf{x}^{(A)}) &= \mathbf{0}. \end{aligned}$$

Voltage Interpolation A first approach is to integrate only the internal part of the active subcircuit at the fine time-grid. Then we get the following active circuit model for $\tilde{\mathbf{x}}^{(A)}$

$$\frac{d}{dt} [\tilde{\mathbf{q}}^{(A)}(t, \mathbf{x}^{(A)})] + \check{\mathbf{j}}^{(A)}(t, \mathbf{x}^{(A)}) = \mathbf{0}, \quad \mathbf{x}^{(A)} = \begin{bmatrix} \tilde{\mathbf{x}}^{(A)} \\ \tilde{\mathbf{x}}^{(A)} \end{bmatrix}. \quad (5)$$

In practise $\tilde{\mathbf{x}}^{(A)}$ will also behave latently, and in this case it is preferable to use voltage interpolation of the terminal voltages $\tilde{\mathbf{x}}^{(A)}$. From the hierarchical linear solver in Pstar (the in-house analogue circuit simulator provided by NXP Semiconductors) [1] we know that (5) is solvable for $\tilde{\mathbf{x}}^{(A)}$. However, stability is now not automatically preserved from the original model. Furthermore the DAE-index can be larger than one, which typically leads to sawtooth-like shapes of $\tilde{\mathbf{x}}^{(A)}$.

Current Interpolation A second approach is to integrate the complete active subcircuit (i.e. using $\mathbf{q}^{(A)}$ rather than $\tilde{\mathbf{q}}^{(A)}$, etc) at the fine time-grid. Then we get the following active circuit model for $\mathbf{x}^{(A)}$

$$\frac{d}{dt} [\mathbf{q}^{(A)}(t, \mathbf{x}^{(A)})] + \mathbf{j}^{(A)}(t, \mathbf{x}^{(A)}) = -\hat{\mathbf{B}}_{(A)} \hat{\mathbf{B}}_{(L)}^T \left[\frac{d}{dt} [\hat{\mathbf{q}}^{(L)}(t, \mathbf{x}^{(L)})] + \hat{\mathbf{j}}^{(L)}(t, \mathbf{x}^{(L)}) \right]. \quad (6)$$

This leads to a more stable situation including the conservation of Kirchhoff's Current Law at the terminals and preservation of the DAE-index. In this case it is preferable to interpolate the terminal currents

$$\mathbf{i}_{L \rightarrow A} = -\hat{\mathbf{B}}_{(A)} \hat{\mathbf{B}}_{(L)}^T \left[\frac{d}{dt} [\hat{\mathbf{q}}^{(L)}(t, \mathbf{x}^{(L)})] + \hat{\mathbf{j}}^{(L)}(t, \mathbf{x}^{(L)}) \right]. \quad (7)$$

This can be done by adding $\mathbf{i}_{L \rightarrow A}$ as unknown or by calculating it explicitly. Direct interpolation of the slow voltages $\mathbf{x}^{(L)}$ is not attractive because often $d_L \gg d_A$. In Pstar for each subcircuit the corresponding terminal current $\frac{d}{dt} [\hat{\mathbf{q}}^{(i)}(t, \mathbf{x}^{(i)})] + \hat{\mathbf{j}}^{(i)}(t, \mathbf{x}^{(i)})$ is stored. Then the vector $\mathbf{i}_{L \rightarrow A}$ can be constructed for each multirate-partitioning.

4 Implicit Interpolation

Interpolation of the currents $\mathbf{i}_{L \rightarrow A}$ causes solvability problems for the active part if the active subcircuits are not grounded (so one may have to ground the most latent coupled terminal unknown). Stability and the differential index are only preserved if all subcircuits are stable DAEs of index one. In general

this property can not be assumed for a circuit simulator. An alternative could be a modified BDF multirate algorithm with implicit interpolation.

In (7) we already introduced the terminal current $\mathbf{i}_{L \rightarrow A}$ from latent-to-active. We also introduce the terminal current $\mathbf{i}_{A \rightarrow L}$ from active-to-latent. Then one can also write the hierarchical circuit model of (6) like

$$\left\{ \begin{array}{l} \frac{d}{dt}[\mathbf{q}^{(L)}(t, \mathbf{x}^{(L)})] + \mathbf{j}^{(L)}(t, \mathbf{x}^{(L)}) = \mathbf{i}_{A \rightarrow L}, \quad i \\ \mathbf{i}_{A \rightarrow L} = -\hat{\mathbf{B}}_{(L)} \hat{\mathbf{B}}_{(A)}^T \left[\frac{d}{dt}[\hat{\mathbf{q}}^{(A)}(t, \mathbf{x}^{(A)})] + \hat{\mathbf{j}}^{(A)}(t, \mathbf{x}^{(A)}) \right], \quad ii \\ \mathbf{i}_{L \rightarrow A} = -\hat{\mathbf{B}}_{(A)} \hat{\mathbf{B}}_{(L)}^T \left[\frac{d}{dt}[\hat{\mathbf{q}}^{(L)}(t, \mathbf{x}^{(L)})] + \hat{\mathbf{j}}^{(L)}(t, \mathbf{x}^{(L)}) \right], \quad iii \\ \frac{d}{dt}[\mathbf{q}^{(A)}(t, \mathbf{x}^{(A)})] + \mathbf{j}^{(A)}(t, \mathbf{x}^{(A)}) = \mathbf{i}_{L \rightarrow A}. \quad iv \end{array} \right. \quad (8)$$

Here $\mathbf{i}_{A \rightarrow L}$ and $\mathbf{i}_{L \rightarrow A}$ are the terminal currents that couple both subcircuits. If the vector $\mathbf{i}_{L \rightarrow A}$ is given it is possible to perform the refinement for $\mathbf{x}^{(A)}$. For the Slow-Fast multirate method $\mathbf{i}_{L \rightarrow A}$ is approximated at the coarse time-grid, based on $\mathbf{x}^{(L)}$. However, it is also possible to approximate $\mathbf{i}_{L \rightarrow A}$ by a different approach. Note that $\mathbf{i}_{L \rightarrow A}$ and $\mathbf{i}_{A \rightarrow L}$ are related by the Kirchhoff's Current Law

$$\hat{\mathbf{B}}_{(L)}^T \mathbf{i}_{A \rightarrow L} + \hat{\mathbf{B}}_{(A)}^T \mathbf{i}_{L \rightarrow A} = \mathbf{0}.$$

Variant I Let us discretise (8i) by Euler Backward with step $H_{n,m} = t_{n-1,m} - t_{n-1,0}$, where $t_{n-1,0} = T_{n-1}$

$$\mathbf{q}^{(L)}(t_{n-1,m}, \mathbf{x}_{n-1,m}^{(L)}) - \mathbf{q}^{(L)}(t_{n-1,0}, \mathbf{x}_{n-1,0}^{(L)}) + H_{n,m} \mathbf{j}^{(L)}(t_{n-1,m}, \mathbf{x}_{n-1,m}^{(L)}) = H_{n,m} \mathbf{i}_{A \rightarrow L}(t_{n-1,m}). \quad (9)$$

We assume that just one Newton step is needed to correct the prediction $\mathbf{y}_{n-1,m}^{(L)}$ of $\mathbf{x}_{n-1,m}^{(L)}$, which is acceptable if $\mathbf{x}^{(L)}$ behaves latently. Thus

$$\mathbf{J}_{n-1,m}^{(L)} (\mathbf{x}_{n-1,m}^{(L)} - \mathbf{y}_{n-1,m}^{(L)}) = H_{n,m} \mathbf{i}_{A \rightarrow L}(t_{n-1,m}) - \mathbf{f}_{n-1,m}^{(L)}, \text{ where}$$

$\mathbf{J}_{n-1,m}^{(L)} = \mathbf{C}^{(L)}(t_{n-1,m}, \mathbf{y}_{n-1,m}^{(L)}) - \mathbf{C}^{(L)}(t_{n-1,0}, \mathbf{x}_{n-1,0}^{(L)}) + H_{n,m} \mathbf{G}^{(L)}(t_{n-1,m}, \mathbf{y}_{n-1,m}^{(L)})$, $\mathbf{f}_{n-1,m} = \mathbf{q}^{(L)}(t_{n-1,m}, \mathbf{y}_{n-1,m}^{(L)}) - \mathbf{q}^{(L)}(t_{n-1,0}, \mathbf{x}_{n-1,0}^{(L)}) + H_{n,m} \mathbf{j}^{(L)}(t_{n-1,m}, \mathbf{y}_{n-1,m}^{(L)})$, $\mathbf{C}^{(L)} = \frac{\partial \mathbf{q}^{(L)}}{\partial \mathbf{x}^{(L)}}$, and $\mathbf{G}^{(L)} = \frac{\partial \mathbf{j}^{(L)}}{\partial \mathbf{x}^{(L)}}$. The matrix $\mathbf{J}_{n-1,m}^{(L)}$ is invertible if the latent part (8i) is solvable, which is a reasonable assumption. Hence

$$\mathbf{x}_{n-1,m}^{(L)} = \mathbf{y}_{n-1,m}^{(L)} + \mathbf{J}_{n-1,m}^{-1} (H_{n,m} \mathbf{i}_{A \rightarrow L}(t_{n-1,m}) - \mathbf{f}_{n-1,m}).$$

We do not want to compute $\mathbf{J}_{n-1,m}^{-1} \mathbf{f}_{n-1,m}$ for all m , so we use linear interpolation of $\mathbf{J}_{n-1,0}^{-1} \mathbf{f}_{n-1,0} = \mathbf{J}_{n-1,0}^{-1} \mathbf{f}(\mathbf{x}_{n-1,0}^{(L)})$ and $\mathbf{J}_{n,0}^{-1} \mathbf{f}_{n,0} = \mathbf{J}_{n,0}^{-1} \mathbf{f}(\mathbf{x}_{n,0}^{(L)})$. Thus we obtain

$$\mathbf{J}_{n-1,m}^{-1} \mathbf{f}_{n-1,m} := \lambda_m \mathbf{J}_{n-1,0}^{-1} \mathbf{f}_{n-1,0} + \mu_m \mathbf{J}_{n,0}^{-1} \mathbf{f}_{n,0}.$$

Here $\lambda_m = 1 - \frac{m}{q}$ and $\mu_m = 1 - \lambda_m = \frac{m}{q}$. For $m = 0$ we have that $\mathbf{x}_{n-1,m}^{(L)} = \mathbf{y}_{n-1,0}^{(L)}$ solves (9), hence we have that $\mathbf{f}_{n-1,0} = \mathbf{0}$. After interpolating the operator \mathbf{J}^{-1} applied to $H_{n,m}\mathbf{i}_{A \rightarrow L}(t_{n-1,m})$, we can approximate $\mathbf{x}_{n-1,m}^{(L)}$ by:

$$\begin{aligned} \mathbf{x}_{n-1,m}^{(L)} &\approx \mathbf{y}_{n-1,m}^{(L)} + (\lambda_m \mathbf{J}_{n-1,0}^{-1} + \mu_m \mathbf{J}_{n,0}^{-1}) H_{n,m} \mathbf{i}_{A \rightarrow L}(t_{n-1,m}) - \mu_m \mathbf{J}_{n,0}^{-1} \mathbf{f}_{n,0} \\ &\approx \mathbf{a}(t_{n-1,m}) + \mathbf{A}(t_{n-1,m}) \mathbf{i}_{A \rightarrow L}(t_{n-1,m}), \end{aligned} \quad (10)$$

where $\mathbf{a}(t_{n-1,m}) = \mathbf{y}_{n-1,m}^{(L)} - \mu_m \mathbf{J}_{n,0}^{-1} \mathbf{f}_{n,0}$ and $\mathbf{A}(t_{n-1,m}) = (\lambda_m \mathbf{J}_{n-1,0}^{-1} + \mu_m \mathbf{J}_{n,0}^{-1}) H_{n,m}$. Hence, one can write:

$$\left\{ \begin{array}{l} \mathbf{i}_{A \rightarrow L} = -\hat{\mathbf{B}}_{(L)} \hat{\mathbf{B}}_{(A)}^T \left[\frac{d}{dt} [\hat{\mathbf{q}}^{(A)}(t, \mathbf{x}^{(A)})] + \hat{\mathbf{j}}^{(A)}(t, \mathbf{x}^{(A)}) \right], \\ \mathbf{i}_{L \rightarrow A} = -\hat{\mathbf{B}}_{(A)} \hat{\mathbf{B}}_{(L)}^T \left[\frac{d}{dt} [\hat{\mathbf{q}}^{(L)}(t, \mathbf{a}(t) + \mathbf{A}(t) \mathbf{i}_{A \rightarrow L})] + \hat{\mathbf{j}}^{(L)}(t, \mathbf{a}(t) + \mathbf{A}(t) \mathbf{i}_{A \rightarrow L}) \right], \\ \frac{d}{dt} [\mathbf{q}^{(A)}(t, \mathbf{x}^{(A)})] + \mathbf{j}^{(A)}(t, \mathbf{x}^{(A)}) = \mathbf{i}_{L \rightarrow A}. \end{array} \right. \quad (11)$$

Next we can compute $\mathbf{x}^{(L)}$ by evaluating formula (10). Thus we get a multirate method of Fastest First type instead of Slowest First type. In contrast to the Compound-Fast multirate method we do not need a compound step now to predict $\mathbf{i}_{L \rightarrow A}$.

Variant II In a similar way as for $\mathbf{x}^{(L)}$, $\mathbf{i}_{L \rightarrow A}$ satisfies

$$\begin{aligned} \mathbf{i}_{L \rightarrow A}(t_{n-1,m}) &= -\hat{\mathbf{B}}_{(A)} \hat{\mathbf{B}}_{(L)}^T \left[\frac{d}{dt} [\hat{\mathbf{q}}^{(L)}(t_{n-1,m}, \mathbf{x}_{n-1,m}^{(L)})] + \hat{\mathbf{j}}^{(L)}(t_{n-1,m}, \mathbf{x}_{n-1,m}^{(L)}) \right] \\ &= -\hat{\mathbf{P}} \left[\frac{d}{dt} [\mathbf{q}^{(L)}(t_{n-1,m}, \mathbf{x}_{n-1,m}^{(L)})] + \mathbf{j}^{(L)}(t_{n-1,m}, \mathbf{x}_{n-1,m}^{(L)}) \right], \end{aligned}$$

where $\hat{\mathbf{P}} = \hat{\mathbf{B}}_{(A)} \hat{\mathbf{B}}_{(L)}^T \left[\hat{\mathbf{B}}_{(L)} \mathbf{0} \right]$. In a similar way as for $\mathbf{x}^{(L)}$ in (10) we can derive the following feedback law for $\mathbf{i}_{L \rightarrow A}$:

$$\mathbf{i}_{L \rightarrow A}(t) \approx \mathbf{b}(t) + \mathbf{B}(t) \mathbf{x}^{(L)}(t). \quad (12)$$

Expressing $\mathbf{x}^{(L)}(t)$ as in (10) we can derive $\mathbf{c}(t)$ and $\mathbf{C}(t)$ such that

$$\mathbf{i}_{L \rightarrow A}(t) \approx \mathbf{c}(t) + \mathbf{C}(t) \mathbf{i}_{A \rightarrow L}(t).$$

This reduces the system (8) even further to the following system for $\mathbf{i}_{A \rightarrow L}$, $\mathbf{x}^{(A)}$:

$$\left\{ \begin{array}{l} \mathbf{i}_{A \rightarrow L} = -\hat{\mathbf{B}}_{(L)} \hat{\mathbf{B}}_{(A)}^T \left[\frac{d}{dt} [\hat{\mathbf{q}}^{(A)}(t, \mathbf{x}^{(A)})] + \hat{\mathbf{j}}^{(A)}(t, \mathbf{x}^{(A)}) \right], \\ \frac{d}{dt} [\mathbf{q}^{(A)}(t, \mathbf{x}^{(A)})] + \mathbf{j}^{(A)}(t, \mathbf{x}^{(A)}) = \mathbf{c}(t) + \mathbf{C}(t) \mathbf{i}_{A \rightarrow L}. \end{array} \right. \quad (13)$$

We can eliminate $\mathbf{i}_{A \rightarrow L}$, which results in the following system for $\mathbf{x}^{(A)}$

$$\frac{d}{dt} [\mathbf{q}^{(A)}(t, \mathbf{x}^{(A)})] + \mathbf{j}^{(A)}(t, \mathbf{x}^{(A)}) = \mathbf{c}(t) - \mathbf{C}(t) \hat{\mathbf{B}}_{(L)} \hat{\mathbf{B}}_{(A)}^T \left[\frac{d}{dt} [\hat{\mathbf{q}}^{(A)}(t, \mathbf{x}^{(A)})] + \hat{\mathbf{j}}^{(A)}(t, \mathbf{x}^{(A)}) \right]. \quad (14)$$

From its structure it can be seen that only the terminal active equations that are directly coupled to the latent part are modified. In fact they are multiplied by a linear transformation. This linear transformation is such that the dynamical behaviour of the original system has been preserved. Again, the vector-valued function $\mathbf{c}(t)$ is an interpolation-based current source.

Variant III In a similar way as for $\mathbf{i}_{L \rightarrow A}$ we can derive the formula

$$\mathbf{i}_{A \rightarrow L}(t) \approx \mathbf{f}(t) + \mathbf{F}(t)\mathbf{x}^{(A)}(t). \quad (15)$$

Combining all three formulae (10), (12) and (15) enables us to express $\mathbf{i}_{L \rightarrow A}$ directly in terms of $\mathbf{x}^{(A)}$:

$$\mathbf{i}_{L \rightarrow A}(t) = \mathbf{g}(t) + \mathbf{G}(t)\mathbf{x}^{(A)}(t).$$

Then we get the following system for $\mathbf{x}^{(A)}$:

$$\left\{ \frac{d}{dt}[\mathbf{q}^{(A)}(t, \mathbf{x}^{(A)})] + \mathbf{j}^{(A)}(t, \mathbf{x}^{(A)}) = \mathbf{g}(t) + \mathbf{G}(t)\mathbf{x}^{(A)}. \quad (16) \right.$$

5 Conclusions

We described a multirate method for hierarchical IC models. It is analysed and tested in [5, 7, 8]. For IC models with many slowly-varying unknowns it is possible to achieve a good speed-up while the accuracy is maintained. We also proposed a new implicit type of interpolation that can solve some typical problems with solvability and stability for the active part. Variant I needs to evaluate all terminal equations for the slow models and solves all terminal currents, which leads to a second order system and can be expensive. But it can also be applied for fast terminal currents $\mathbf{i}_{L \rightarrow A}$, $\mathbf{i}_{A \rightarrow L}$. Variant II only needs to evaluate active elements but it still needs $\mathbf{i}_{A \rightarrow L}$ as additional unknown. Therefore it still can be applied for active $\mathbf{i}_{A \rightarrow L}$. Variant III really reduces to a system for only the active part. It is only allowed if all terminal currents behave slowly. For the third variant it is clear that $\mathbf{i}_{L \rightarrow A}$ is replaced by a combination of current sources and resistors. In fact this is model reduction of the large latent part.

References

1. Fijnvandraat, J.G., Houben, S.H.M.J., ter Maten, E.J.W., Peters, J.M.F.: Time domain analog circuit simulation. *J. Comput. Appl. Math.* **185**(2), 441–459 (2006)
2. ter Maten, J., Verhoeven, A., El Guennoui, A., Beelen, Th.: Multirate hierarchical time integration for electronic circuits. In: PAMM, Virtual Annual Meeting Proceedings of the GAMM conference, pp. 819–820. Luxembourg (2005)
3. Savcenko, V.: Multirate Numerical Integration for Ordinary Differential Equations. PhD thesis, CWI, Amsterdam (2008)
4. Striebel, M.: Hierarchical Mixed Multirating for Distributed Integration of DAE Network Equations in Chip Design. PhD thesis, Bergische University of Wuppertal, Wuppertal, Germany (2006)

5. Verhoeven, A.: Redundancy Reduction of IC Models by Multirate Time Integration and Model Order Reduction. PhD thesis, Eindhoven University of Technology, Department of Mathematics and Computer Science, Eindhoven (2008)
6. Verhoeven, A., El Guennouni, A., ter Maten, E.J.W., Mattheij, R.M.M.: A general compound multirate method for circuit simulation problems. In: Anile, A.M., Ali, G., Mascali, G. (eds.) *Scientific Computing in Electrical Engineering*, pp. 143–150. Capo d’Orlando, Italy. Springer, Berlin (2006)
7. Verhoeven, A., ter Maten, E.J.W., Mattheij, R.M.M., Tasić, B.: Stability analysis of the BDF slowest first multirate methods. *Int. J. Comput. Math.* **84**, 895–923 (2007)
8. Verhoeven, A., Tasić, B., Beelen, T.G.J., ter Maten, E.J.W., Mattheij, R.M.M.: BDF Compound-Fast multirate transient analysis with adaptive stepsize control. *J. Num. Anal. Ind. Appl. Math.* **3**(3–4) (2008)