Aspects of multirate time integration methods in circuit simulation problems

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Computer Science

CASA-Report 04-26
October 2004

Aspects of multirate time integration methods in circuit simulation problems

by

A. El Guennouni, A. Verhoeven,
E.J.W. ter Maten and T.G.J. Beelen

Centre for Analysis, Scientific computing and Applications
Department of Mathematics and Computer Science
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven, The Netherlands
ISSN: 0926-4507

Aspects of Multirate Time Integration Methods in Circuit Simulation Problems

A. El Guennouni[1,3], A. Verhoeven[2], E.J.W. ter Maten[2,3], and T.G.J Beelen[3]

[1] Yacht, Eindhoven, The Netherlands
[2] Eindhoven University of Technology, Eindhoven, The Netherlands
[3] Philips Research, Eindhoven, The Netherlands. Jan.ter.Maten@philips.com

1 Introduction

We present a new robust compound step strategy for multirate time integration methods. The strategy applies to DAEs and to time-dependent PDEs as well. We will compare this to other known strategies like Slowest First and Fastest First and modern ones that involve some form of a compound step integration during the process.
The circuit equations can be written as a DAE system of equations

$$\frac{d}{dt}[\mathbf{q}(t,\mathbf{x})] + \mathbf{j}(t,\mathbf{x}) = 0. \tag{1}$$

We assume that some partitioning is given that distinguishes between "slow" and "fast" varying components $\mathbf{x}_S$ and $\mathbf{x}_F$. In this case (1) can be written as

$$\frac{d}{dt}[\mathbf{q}_F(t,\mathbf{x}_F,\mathbf{x}_S)] + \mathbf{j}_F(t,\mathbf{x}_F,\mathbf{x}_S) = 0 \tag{2}$$

$$\frac{d}{dt}[\mathbf{q}_S(t,\mathbf{x}_F,\mathbf{x}_S)] + \mathbf{j}_S(t,\mathbf{x}_F,\mathbf{x}_S) = 0 \tag{3}$$

Ordinary multirate time integration aims to integrate (2)-(3) to a same accuracy using different time-steps H and h, in which $H \gg h$. We intend to apply the approach to mixed signal simulation in which digital and analog circuitry are combined. The digital part often shows latent time behaviour, while the analog part often shows time varying activity. In addition, on the digital part less accuracy is needed than on the analog part. This gives way to combine multirate time integration with distributed tolerances.
Because circuit simulators usually apply Backward Differentiation Formula (BDF) methods as time integrator we will consider multirate time integration for the most simple one, the Euler Backward method.
The Slowest First strategy integrates first (2) for $\mathbf{x}_S$ using extrapolated values for $\mathbf{x}_F$ and step-size H. Next (3) is integrated repeatedly for $\mathbf{x}_F$ using

interpolated values of $\mathbf{x}_S$ and step-size h.
The Fastest First strategy simply starts with (3). The first approach benefits from being better suited when dealing with automatic step-size control mechanism. However both methods have weak stability properties, due to the extrapolation involved [2]. For this reason we were led to study implicit methods. Interesting ones can be cast in the following General Compound Strategy, in which $q = \frac{H}{h}$ and $0 < \alpha \leq 1$ is just a parameter:

ALGORITHM 1 *A General Compound (G.C.) Strategy*

Compound phase: *Solve for* $\mathbf{x}_S^{n+q}$ *and* $\mathbf{x}_F^{n+\alpha q}$:

$$\mathbf{q}_F(\mathbf{x}_F^{n+\alpha q}, \hat{\mathbf{x}}_S^{\alpha}) - \mathbf{q}_F(\mathbf{x}_F^n, \mathbf{x}_S^n) + \alpha H \mathbf{j}_F(\mathbf{x}_F^{n+\alpha q}, \hat{\mathbf{x}}_S^{\alpha}) = 0 \tag{4}$$

$$\hat{\mathbf{x}}_S^{\alpha} - \mathbf{x}_S^n - \alpha(\mathbf{x}_S^{n+q} - \mathbf{x}_S^n) = 0 \tag{5}$$

$$\mathbf{q}_S(\hat{\mathbf{x}}_F^{\alpha}, \mathbf{x}_S^{n+q}) - \mathbf{q}_S(\mathbf{x}_F^n, \mathbf{x}_S^n) + H \mathbf{j}_S(\hat{\mathbf{x}}_F^{\alpha}, \mathbf{x}_S^{n+q}) = 0 \tag{6}$$

$$\hat{\mathbf{x}}_F^{\alpha} - \mathbf{x}_F^n - \frac{1}{\alpha}(\mathbf{x}_F^{n+\alpha q} - \mathbf{x}_F^n) = 0 \tag{7}$$

Refinement phase: *Solve for* $\mathbf{x}_F^{n+j+1}$ $(j = 0, \ldots, q-1)$:

$$\mathbf{q}_F(\mathbf{x}_F^{n+j+1}, \hat{\mathbf{x}}_S^{n+j+1}) - \mathbf{q}_F(\mathbf{x}_F^{n+j}, \hat{\mathbf{x}}_S^{n+j}) + h \mathbf{j}_F(\mathbf{x}_S^{n+j+1}, \hat{\mathbf{x}}_S^{n+j+1}) = 0 \tag{8}$$

$$\hat{\mathbf{x}}_S^{n+j+1} - \mathbf{x}_S^n + \frac{j+1}{q}(\mathbf{x}_S^{n+q} - \mathbf{x}_S^n) = 0 \tag{9}$$

Here (4)-(7) form a "Compound Step" in which $\mathbf{x}_S$ is determined at $t = t_n + qh = t_n + H$, together with implicitly determined $\mathbf{x}_F$. If $\alpha = 1$, this "Compound Step" is just the result of Euler Backward with a large step H, which is easy to implement. If $\alpha = \frac{1}{q}$, the solutions $\mathbf{x}_S^{n+q}$ and $\mathbf{x}_F^{n+1}$ are simultaneously calculated. This option corresponds to the multirate method described in [1, 3] (but for Runge-Kutta and Rosenbrock-Wanner methods).
Integration of (8) is the "Refinement Step" for the fast part. It uses interpolated values $\hat{\mathbf{x}}_S^{n+j+1}$ as expressed in (9).
Clearly the GC step methods solve a larger system during the Compound Step phase than in the Slowest First or the Fastest First strategies. However, in most mixed signal applications the size of the digital parts exceeds that of the analog part several times. The GC step methods have much better stability properties than the Slowest First or the Fastest First strategies as conjectured by [4] and which is proved in [5]. For instance, considering the next two-dimensional test equation

$$\begin{pmatrix} \dot{x}_F \\ \dot{x}_S \end{pmatrix} = \underbrace{\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}}_{\mathbf{A}} \begin{pmatrix} x_F \\ x_S \end{pmatrix} \tag{10}$$

the following stability conditions for $\mathbf{A}$ are derived in [5]:

SF	GC	GC ($\alpha = 1$)
$a_{11} < 0$	$a_{11} < 0$	$a_{11} < 0$
$a_{22} < 0$	$\alpha a_{11} + a_{22} < 0$	$a_{11} + a_{22} < 0$
$\lvert a_{12}a_{21}\rvert < \lvert a_{11}a_{22}\rvert$	$-a_{11}a_{22} - 2\alpha a_{11}^2 < a_{12}a_{21}$	$-a_{11}a_{22} - 2a_{11}^2 < a_{12}a_{21}$
	$a_{12}a_{21} < a_{11}a_{22}$	$a_{12}a_{21} < a_{11}a_{22}$

In the sequel we assume $\alpha = 1$, which is the most robust choice. We will demonstrate that this strategy elegantly fits hierarchical circuit definition. Furthermore, the impact of the partionining will be considered.

2 Model Problem

Using Modified Nodal Analysis, in circuit simulation applications $\mathbf{q}(t, \mathbf{x}) = \sum_e \mathbf{B}_e q_e(t, \mathbf{B}_e^T \mathbf{x})$, in which q_e is a local branch function. For instance, for a (linear) capacitor $C(a, b)$ between nodes a and b, $q_e = C$ and $\mathbf{B}_e = \mathbf{e}_a - \mathbf{e}_b$, in which $\mathbf{e}_a$ (resp $\mathbf{e}_b$) is a canonical unit vector with a 1 at place a (resp. b), and zeros elsewhere. The operators $\mathbf{B}_e$ are defined by the topology of the network. They do not depend on t or on $\mathbf{x}$. Similar results hold for the function $\mathbf{j}$. Assembly can be grouped to sub-circuits to fit hierarchical circuit definitions. In this way

$$\mathbf{q}(t, \mathbf{x}) = \sum_s \mathbf{B}_s \mathbf{q}_s(t, \mathbf{B}_s^T \mathbf{x})$$

where $\mathbf{B}_s^T \mathbf{x}$ selects the unknowns at the sub-circuit level and

$$\mathbf{q}_s(t, \mathbf{y}) = \sum_e \mathbf{B}_e q_e(t, \mathbf{B}_e^T \mathbf{y})$$

defines the assembly of $\mathbf{q}_s$ inside the sub-circuit.

A simple model problem is shown in Fig. 1. With $\mathbf{x} = (V_1, V_2, i_E, V_3, V_4)^T$, the functions $\mathbf{q}$ and $\mathbf{j}$ are given by

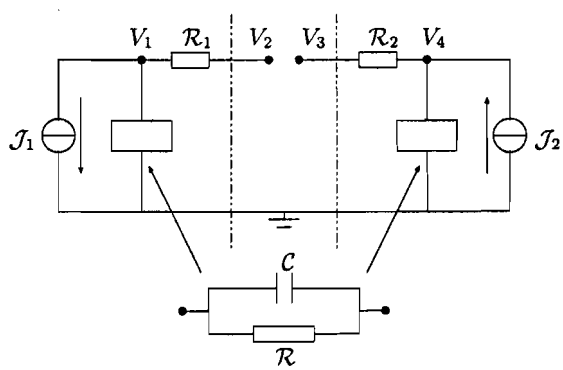

Fig. 1. Simple model circuit; Fast at the left, Slow at the right

$$\mathbf{q}(t,x) = \begin{bmatrix} \mathcal{C}\mathbf{x}_1 \\ 0 \\ 0 \\ 0 \\ \mathcal{C}\mathbf{x}_5 \end{bmatrix}, \; J(t,x) = \begin{bmatrix} \mathcal{J}_1(t,\mathbf{x}_1) - \frac{(\mathbf{x}_1-\mathbf{x}_2)}{\mathcal{R}_1} \\ \frac{(\mathbf{x}_1-\mathbf{x}_2)}{\mathcal{R}_1} - \mathbf{x}_3 \\ \mathbf{x}_2 - \mathbf{x}_4 \\ \frac{(\mathbf{x}_4-\mathbf{x}_5)}{\mathcal{R}_2} - \mathbf{x}_3 \\ \mathcal{J}_2(t,\mathbf{x}_5) - \frac{(\mathbf{x}_5-\mathbf{x}_4)}{\mathcal{R}_2} \end{bmatrix} \tag{11}$$

The source functions $\mathcal{J}_k(t,x)$ may be defined by $\mathcal{J}_k(t,x) = \sin(\omega_k t)$ with $\omega_1 \gg \omega_2$. The role of i_E of the short E serves to explicitly obtain the terminal current between the fast and the slow part (the short E can be included automatically as a "virtual" glue-elements by the simulator, see also section 3). In the refinement phase a current source can be used to define the outgoing current of the slow part.
Several partitionings $\mathcal{P}_k$ may be considered. Here the following are considered:

- $\mathcal{P}_1$: $x_f = [V_1, V_2, i_E, V_3]$ and $x_s = [V_4]$.
- $\mathcal{P}_2$: $x_f = [V_1, V_2, i_E]$ and $x_s = [V_3, V_4]$.

If a partitioning $\mathcal{P}$ is chosen, topological matrices $\mathbf{B}_f$ and $\mathbf{B}_s$ can be defined in the same style as before to define for instance $q(t, \mathbf{B}_s\mathbf{x})$.
The Euler Backward Compound method ($\alpha = 1$) proved to be very stable due to the implicit extrapolation. Considering the last method more closely for different partitionings, we observe that $\mathcal{P}_1$ performs best. Results are shown in Fig. 2. However, despite the less accurate results, due to the interpolation

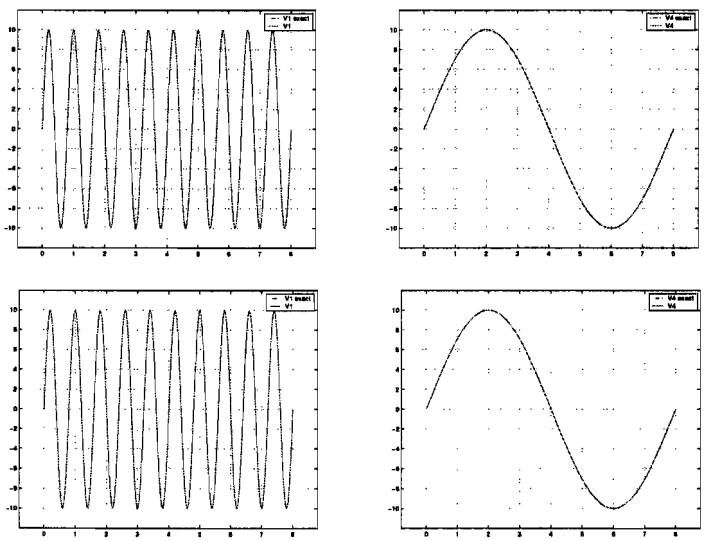

Fig. 2. Results with Partitioning $\mathcal{P}_1$ (at the top) and Partitioning $\mathcal{P}_2$ (at the bottom). $\mathcal{R}_i = 10^5, \mathcal{R} = 10, \mathcal{C} = 2.10^{-10}, H = 0.16$ and $h = 0.032$. $Error(\mathcal{P}_1(V_1)) \approx Error(\mathcal{P}_2)(V_1) \approx 10^{-10}$. $Error(\mathcal{P}_1(V_2)) = 9.63\ 10^{-7} < Error(\mathcal{P}_2(V4)) = 8.92\ 10^{-5}$

error, partitioning $\mathcal{P}_2$ is very attractive, because it elegantly fits an existing hierarchical evaluator: the main part of the partitioning is along the boundary of the known sub-circuits S_1 and S_2 and thus this partitioning may even be given by the user. In the algorithm S_1 and S_2 are treated as in ordinary transient simulation (with their own step-size).

3 Interface treatment fitting hierarchical sub-circuits

At the interface, the partitioning concentrates in the "glue-elements" . A more general glue-elements is shown in Fig. 3, from which it is clear how it can be generated. For instance, it applies to sub-circuits of which two terminals of one are connected to the same terminal of another. Note also that the connections between the sub-circuits are treated more symmetrically. Now each short can measure a particular terminal current. In the glue-elements this facilitates particular implementations for multirate integration, also when dealing with partitionings with more different multirate behaviour.

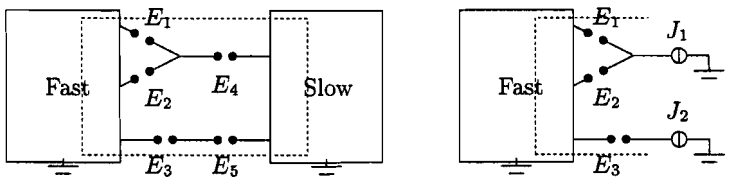

Fig. 3. Generation and different role of glue-elements. In the compound phase (at the left) shorts E_4 and E_5 allow to measure currents at the slow boundary. In the refinement phase (at the right) these current values are used to define the current sources J_1 and J_2. The boxes "fast" and "slow" can be treated as black boxes.

References

1. A. Bartel and M. Günther. A multirate w-method for electrical networks in state-space formulation. *J. of Comput. and Applied Maths.*, 147:411–425, 2002.
2. C.W. Gear and D.R. Wells. Multirate linear multistep methods. *BIT*, 24:484–502, 1984.
3. M. Günther, A. Kværnø, and P. Rentrop. Multirate partitioned runge-kutta methods. *BIT*, 41:504–515, 2001.
4. S. Skelboe. Accuracy of decoupled implicit integration. *SIAM J. Sc. Comput.*, 21(6):2206–2204, 2000.
5. A. Verhoeven, A. El Guennouni, , E.J.W. ter Maten, and R.M.M. Mattheij. A general compound multirate method for circuit simulation problems. Presented at *Scientific Computing in Electrical Engineering*, Capo D'Orlando, Sicily, Italy, 2004.