NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET

Multirate Methods in Electrical Circuit Simulation

by

Andreas Bartel, Michael Günther and Anne Kværnø

PREPRINT
NUMERICS NO. 2/2001

NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL
http://www.math.ntnu.no/preprint/numerics/2001/N2-2001.ps
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

Multirate Methods in Electrical Circuit Simulation

Andreas Bartel[1], Michael Günther[2] *, and Anne Kværnø[3]

[1] Universität Karlsruhe (TH), IWRMM, D-76128 Karlsruhe, Germany.
[2] Universität Ulm, Fakultät für Mathematik und Wirtschaftswissenschaften, D-89081 Ulm.
[3] Norwegian University of Science and Technology, Dept. of Mathematical Sciences, N-7491 Trondheim, Norway.

Abstract. Multirate methods in the simulation of coupled systems adapt the numerical effort to the activity level of the respective subsystems. Here, two different approaches will be presented: one based on operator splitting and a second using the concept of generalised multirated. For both the inverter-chain-benchmark serves as a test set, which will confirm the potential of multirate methods.

1 Introduction

The dynamical behaviour of coupled systems is often characterised by large varying time scales of subsystems. In radio frequency applications, for example, one has to investigate coupled systems of analogue and digital networks. Whereas the analogue part runs on the micro second scale, the digital part is measured in nano seconds. In every instant of time, VLSI circuits consist of a large latent part of inactivity plus a very small subsystem where signal processing takes place. This multirate behaviour has to be exploited not only for the sake of efficiency, but also to be empowered to solve even larger systems in an appropriate amount of time.

In the following, we assume that the solution $\mathbf{y}(t)$ of the initial value problem is split into active components $\mathbf{y}_A(t)$ and latent components $\mathbf{y}_L(t)$, thus

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}_A \\ \mathbf{y}_L \end{pmatrix}, \quad \mathbf{y}_A \in \mathbb{R}^{n_A}, \quad \mathbf{y}_L \in \mathbb{R}^{n_L}, \quad n_A + n_L = n.$$

The result is a split system,

$$\begin{aligned} \dot{\mathbf{y}}_A(t) &= \mathbf{f}_A(\mathbf{y}_A, \mathbf{y}_L), \qquad \mathbf{y}_A(t_0) = \mathbf{y}_{A,0}, \\ \dot{\mathbf{y}}_L(t) &= \mathbf{f}_L(\mathbf{y}_A, \mathbf{y}_L), \qquad \mathbf{y}_L(t_0) = \mathbf{y}_{L,0} \end{aligned} \tag{1}$$

which for simplicity, but without loss of generality, is assumed autonomous. Both subsystems are considered stiff. The partitioning might change during

* Permanent address: Universität Karlsruhe (TH), IWRMM, D-76128 Karlsruhe, Germany.

the integration, thus some partitioning strategy is required. The set of active components constitute a small subset of the whole system.

If this system is solved simultaneously, the step size is governed by the dynamics of the active components. In a multirate method, only the active part is integrated using such step sizes, the latent part can be solved by far larger steps. The step size used by the active part is called *micro steps* size, and denoted by h. The step size for the latent part is called *macro steps* size, denoted by H. If we now assume that the work of the integration is proportional to the number of steps times the number of equations, which mirrors time consuming model evaluations in circuit simulation, we find the following rough estimate for the speedup that can be achieved by using a multirate method:

$$S = \frac{n\,m}{n_L + n_A\,m} \approx \frac{1}{\dfrac{1}{m} + \dfrac{n_A}{n}}$$

where $m = H/h$, the average number of micro steps within a macro step. In the last expression, we have assumed that $n_A << n_L$. Thus, if 10% of the components are active and the the time-scales of the two subsystems differ with a factor of 10, the speedup will be about 5. The inevitable overhead of multirate methods has not been taken into account, if so done, the speedup might be reduced significantly. Still, with a high degree of latency (n_A/n small), and if the difference in time scales can be exploited, multirate methods might count for a significant reduction in computational time, as the examples at the end of this paper will confirm.

Multirate methods aim for a blockwise integration of variables using inherent step sizes. Synchronisation and coupling are main issues. In the next sections, two different strategies for multirating will be described, each with its own strengths and weaknesses. The first one is based on the idea of partitioned one-step methods. For problem (1), the general method with error estimate will be described. The basic idea can be used to construct methods for systems working on more than two time scales and for systems of DAEs. Although possible, the actual derivation of such extensions might turn out to be quite cumbersome.

The second idea is based on operator splitting. This yields a lot of flexibility what concerns the choice of underlying methods, and to which problems the method can be applied. However, the method is only of order two, higher order methods exist, but are rather expensive. No reasonable error estimate is currently available.

2 Mixed multirate method

The *Generalised Multirate* method (GM) can be understood as one-step methods with fixed micro steps, [4]; couplings are realised via internal stages similar to the Runge-Kutta idea. The method regards the system on the large macro scale (slower components) and leaves the computation fixed once the micro step size is predicted - in other words, rejection of a micro step means to update the whole macro step as far as computed.

2.1 The basic method

To obtain a more flexible method, the idea of mixed multirate is to relax the fixed micro step character: use the incremental formulation of GM just for the first step in each component (the so-called *compound step*). Furthermore, employ dense output formulae, which give the already computed solution on the finer grid for active components. To state the formulae, sets of coefficients are involved:

latent				compound		active		later micros	
$\mathcal{A}_L$	$\mathcal{G}_L$	$\mathcal{D}_L$	$\mathcal{N}_L$	$\mathcal{A}_A$	$\mathcal{G}_A$	$\mathcal{D}_A$	$\mathcal{N}_A$	$\mathcal{A}$	$\mathcal{G}$
$\mathbf{b}_L(\theta)$		$\mathbf{d}_L$			$\mathbf{b}_A$	$\mathbf{d}_A$		$\mathbf{b}$	$\mathbf{d}$

where $\mathcal{A}$, $\mathcal{G}$ and $\mathbf{b}[(\theta)]$ define the usual coefficients of *Rosenbrock-Wanner* methods (ROW) [6] – possibly with dense output: $\theta \in [0, 1]$; $\mathbf{d}$ are the weights for the embedded method; $\mathcal{D}$ gives the internal stages and $\mathcal{N}$ is counterpart to $\mathcal{G}$. In the formulae below, the entries of the coefficient matrices are corresponding Greek lower case letters. The mixed method solution (for any part) is given by

$$\mathbf{y}_A^h = \mathbf{y}_{A,0} + \sum_{i=1}^{s_A} b_i \mathbf{k}_{A,i}, \qquad \mathbf{y}_L^H = \mathbf{y}_{L,0} + \sum_{i=1}^{s_L} b_{L,i}(1) \mathbf{k}_{L,i},$$

where the increments $\mathbf{k}_{\cdot,i}$ are defined as:

compound step

$$\begin{aligned}\mathbf{k}_{L,i} =& H\,\mathbf{f}_L\Big(\mathbf{y}_{L,0} + \sum_{j=1}^{s_L} \alpha_{ij}^L \mathbf{k}_{L,j},\, \mathbf{y}_{A,0} + m \sum_{j=1}^{s_A} \delta_L \mathbf{k}_{A,j}^0\Big) + H \left.\frac{\partial \mathbf{f}_L}{\partial \mathbf{y}_L}\right|_{\mathbf{y}_0} \sum_{j=1}^{s_L} \gamma_{ij}^L \mathbf{k}_{L,j} \\ &+ m\,H \left.\frac{\partial \mathbf{f}_L}{\partial \mathbf{y}_A}\right|_{\mathbf{y}_0} \sum_{j=1}^{s_A} \nu_{ij}^L \mathbf{k}_A^0 \qquad (2)\end{aligned}$$

$$\begin{aligned}\mathbf{k}_{A,j}^0 =& h\,\mathbf{f}_A\Big(\mathbf{y}_{L,0} + 1/m \sum_{j=1}^{s_L} \delta_{ij}^A \mathbf{k}_{L,j},\, \mathbf{y}_{A,0} + \sum_{i=1}^{s_A} \alpha_{ij}^A \mathbf{k}_{A,j}^0\Big) + h \left.\frac{\partial \mathbf{f}_A}{\partial \mathbf{y}_A}\right|_{\mathbf{y}_0} \sum_{j=1}^{s_A} \gamma_{ij}^A \mathbf{k}_A^0 \\ &+ h/m \left.\frac{\partial \mathbf{f}_A}{\partial \mathbf{y}_L}\right|_{\mathbf{y}_0} \sum_{j=1}^{s_L} \nu_{ij}^A \mathbf{k}_L\,. \qquad (3)\end{aligned}$$

later micro steps: $\chi = h/H$, while $\chi \leq 1$, $\chi = \chi + h/H$

$$\mathbf{k}_{A,i}^{\chi} = h\,\mathbf{f}_A\Big(\mathbf{y}_L^{H,\,\mathrm{ds}}\Big(\chi + \frac{\alpha_i}{m}\Big),\, \mathbf{y}_{A,\chi} + \sum_{j=1}^{s} \alpha_{ij} \mathbf{k}_{A,j}^{\chi}\Big) + h \left.\frac{\partial \mathbf{f}_A}{\partial \mathbf{y}_A}\right|_{\mathbf{y}_\chi} \sum_{j=1}^{s} \gamma_{ij} \mathbf{k}_{A,j}^{\chi} \qquad (4)$$

and the Jacobian (in the later micro steps) is evaluated at

$$\mathbf{y}_\chi = \left(\mathbf{y}_L^{H,\mathrm{ds}}(\chi),\ \mathbf{y}_{A,\chi}^h\right), \quad \text{with} \quad \mathbf{y}_L^{H,\mathrm{ds}}(\chi) = \mathbf{y}_0 + \sum_{i=1}^{s_L} b_i(\chi)\,\mathbf{k}_{L,i}. \tag{5}$$

Usually, an error check is performed just after the computation of the compound step. The scaled error estimate is used, which for the active part is given by

$$\mathrm{err}_A = (\mathcal{I} - \gamma h \mathbf{D}\mathbf{f}_A)^{-1} \sum_i (b_i - d_i)\mathbf{k}_{A,i} \tag{6}$$

and similar for the latent part. Later, the mixed formula for the micro steps allows not only an error check, but also an adaptation of the micro step size and $m\,(= H/h)$ to be rational. At the end of the macro step the partition is evaluated (see below) and step sizes are predicted for each part: eq. (6) is applied to every variable. With the usual safety precautions one obtains the adapted step size for the corresponding component. The minimum in each part yields the corresponding new step size. Applying P-Theory [5], order equations and a coefficient set for a method of order 3, with a second order error estimating method have been exhibited [1].

2.2 Adaptation to Cheap Jacobian

One of the surprising results in the numerics of circuit simulation is the cheap availability of the Jacobian due to data structures: approximately one obtains the Jacobian and the corresponding right-hand side at cost of 1.4 times the cost of the right-hand side only [3]. This can be used to reduce the number of stages while keeping the order and thus minimise the effort. To do so, terms of the type

$$h \sum_{j=1}^{i-1} \sum_{q=1}^{i} \zeta_{iqj} \mathbf{f}'(\mathbf{a}_q)\mathbf{k}_j$$

have to be added to the right-hand sides of the increments. The computation of the order conditions will now involve the investigations of a *partitioned abstract adapted ROW* method, where the above term is taken into account.

2.3 Adaption to charge-oriented DAE network equations

One more or less straight forward adaptation can be made using the concepts of charge oriented ROW methods [2] to the following (autonomous) index-1 setting:

$$\begin{aligned} A_1\dot{\mathbf{y}}_1 &= \mathbf{f}_1(\mathbf{x}_1, \mathbf{x}_2) & \qquad A_2\dot{\mathbf{y}}_2 &= \mathbf{f}_2(\mathbf{x}_1, \mathbf{x}_2) \\ 0 &= \mathbf{y}_1 - \mathbf{q}_1(\mathbf{x}_1) & 0 &= \mathbf{y}_2 - \mathbf{q}_2(\mathbf{x}_2) \end{aligned}$$

together with the index-1 hypothesis: $A_i \frac{\partial \mathbf{q}_i}{\partial \mathbf{x}_i}$ regular, where A_i are rectangular incidence matrices. Note that in this formulation there is no coupling via the algebraic equations. Let $\mathbf{f}$ denote the static part, $A = (A_1\, A_2)$ the topology and $\mathbf{y}$ the charges, then from the point of circuit simulation all couplings are given via static elements and sources. The mixed algorithm has now the from:

compound step

$$\left(\mathbf{A}_1\frac{\mathrm{d}\mathbf{q}_1}{\mathrm{d}\mathbf{y}_1}-\gamma^L h_1\frac{\partial \mathbf{f}_1}{\partial \mathbf{y}_1}\right)\kappa_{1,i}=\mathbf{q}_{1,0}-\mathbf{q}_1(\hat{\mathbf{a}}_{1,i})+h_1\sum_{j=1}^{i-1}\beta_{ij}^L\mathbf{f}_1(\hat{\mathbf{a}}_{1,j},\hat{\mathbf{d}}_{1,j})+h_1\frac{\partial \mathbf{f}_1}{\partial \mathbf{y}_1}\sum_{j=1}^{i-1}\beta_{ij}^L\kappa_{1,j}$$

$$\left(\mathbf{A}_2\frac{\mathrm{d}\mathbf{q}_2}{\mathrm{d}\mathbf{y}_2}-\gamma^A h_2\frac{\partial \mathbf{f}_2}{\partial \mathbf{y}_2}\right)\kappa_{2,i}=\mathbf{q}_{2,0}-\mathbf{q}_2(\hat{\mathbf{a}}_{2,i})+h_2\sum_{j=1}^{i-1}\beta_{ij}^A\mathbf{f}_2(\hat{\mathbf{d}}_{2,j},\hat{\mathbf{a}}_{2,j})+h_2\frac{\partial \mathbf{f}_2}{\partial \mathbf{y}_2}\sum_{j=1}^{i-1}\beta_{ij}^A\kappa_{2,j}$$

$$\hat{\mathbf{a}}_{p,i} = \mathbf{y}_{p,0} + \sum_{j=1}^{i-1}\sigma_{ij}^{(p)}\kappa_{p,j} \qquad \hat{\mathbf{d}}_{p,i} = \mathbf{y}_{\bar{p},0} + \frac{h_{\bar{p}}}{h_p}\sum_{j=1}^{i-1}\rho_{ij}^{(p)}\kappa_{\bar{p},j} \qquad \{p,\,\bar{p}\}=P=\{1,\,2\}$$

later micro steps

$$\left(\mathbf{A}_2\frac{\mathrm{d}\mathbf{q}_2}{\mathrm{d}\mathbf{y}_2}-\gamma h_3\frac{\partial \mathbf{f}_2}{\partial \mathbf{y}_2}\right)\kappa_{2,i}^{\chi}=\mathbf{q}_{2,0}-\mathbf{q}_2(\hat{\mathbf{a}}_{2,i}^{\chi})+h_1\sum_{j=1}^{i-1}\beta_{ij}\mathbf{f}_2(\hat{\mathbf{d}}_{2,j}^{\chi},\hat{\mathbf{a}}_{2,j}^{\chi})+h_3\frac{\partial \mathbf{f}_2}{\partial \mathbf{y}_2}\sum_{j=1}^{i-1}\beta_{ij}\kappa_{2,j}^{\chi}$$

$$\hat{\mathbf{a}}_{2,i}^{\chi}\ \text{(as above: }\hat{\mathbf{a}}_{2,i}) \qquad \hat{\mathbf{d}}_{2,i}^{\chi} = \mathbf{y}_{1,0} + \sum_{j=1}^{s_1} b_j^L\left(\frac{(\chi+\alpha_3)\,h_2}{h_1}\right)\kappa_{1,j}\ (\chi = 1,\ldots,m).$$

Following the approach of [7], one can use the adapted P-theory once again to obtain order conditions.

3 Operator splitting methods

Operator splitting methods are well known in the field of numerical solution of both ordinary and partial differential equations. By decomposing the system into subsystems, each integrated separately, an approximation to the original problem is obtained by composition. How to do the decomposition depends on the application. As an example, in Hamiltonian mechanics the solution vector is split into a momentum and a position vector. Another option is to split the system into a stiff and a non-stiff part. In our case, with fast and slow solution components, the two subsystems will be

$$\mathbf{y}_A' = \mathbf{f}_A(\mathbf{y}_A,\mathbf{y}_L),\ \ \mathbf{y}_L' = \mathbf{0}, \tag{7a}$$

$$\mathbf{y}_A' = \mathbf{0},\ \ \mathbf{y}_L' = \mathbf{f}_L(\mathbf{y}_A,\mathbf{y}_L) \tag{7b}$$

with the solutions

$$\mathbf{y}_A(t) = \phi_A(t-t_0;\mathbf{y}_{A,0},\mathbf{y}_{L,0}), \qquad \mathbf{y}_L(t) = \mathbf{y}_{L,0} \tag{8a}$$

$$\mathbf{y}_L(t) = \phi_L(t-t_0;\mathbf{y}_{A,0},\mathbf{y}_{L,0}), \qquad \mathbf{y}_A(t) = \mathbf{y}_{A,0} \tag{8b}$$

respectively. One step of a second order splitting method, as proposed by Strang [8], will then be

$$\begin{aligned}
\mathbf{y}_{A,1/2} &= \phi_A(H/2; \mathbf{y}_{A,0}, \mathbf{y}_{L,0}) \\
\mathbf{y}_{L,1} &= \phi_L(H; \mathbf{y}_{A,1/2}, \mathbf{y}_{L,0}) \\
\mathbf{y}_{A,1} &= \phi_A(H/2; \mathbf{y}_{A,1/2}, \mathbf{y}_{L,1}).
\end{aligned}$$

In practice, the values $\mathbf{y}_{A,1/2}$, $\mathbf{y}_{L,1}$ and $\mathbf{y}_{A,1}$ have to be computed numerically. In principle, the choice of numerical methods is free, there are however some considerations that might be done. First of all, we will assume that the computation of the latent part (7b) is the most time-consuming part of the algorithm. We will therefore choose the macro step size H so that $\mathbf{y}_{L,1}$ is computed by one step of the underlying method. In that case, the integration of (7b) has to be restarted every step, thus a genuine one-step method will be preferable. The active components $\mathbf{y}_A$ are solved for by several micro steps over the macro step. Still, the integration is restarted in the middle of each macro step, thus both Runge-Kutta and ROW methods are potential schemes for the active part.

The proposed approach is simple and flexible. It can easily be applied to index-1 differential-algebraic equations, and to problems working on more than two time scales. Unfortunately, the order of the method is low, only two, and there is no error control involved. In some situations, the error from the splitting method can be controlled by a careful partitioning strategy, but no general approach is currently available. Higher order method can easily be derived, see [9]. However, a 4th order method requires 3 internal latent steps, a 6th order method 7. In addition, higher order methods for the internal stages are required. Altogether, we do not consider higher order splitting methods as realistic in electrical circuit simulations.

4 Numerical experiments

A MOSFET inverter transforms a given signal from the logical value 0 to 1 and vice versa. A series of coupled inverters delays the runtime of the signal, see Figure 1. A window of active components passes through the circuit in time. Due to its regular behaviour the inverter chain has become popular as a benchmark problem. The mathematical model is given by

$$\begin{aligned}
\dot{U}_1 &= (U_{op} - U_1)/(RC) - K \cdot g(U_{in}, U_1, U_0)/C \\
\dot{U}_k &= (U_{op} - U_k)/(RC) - K \cdot g(U_{k-1}, U_k, U_0)/C, \qquad k = 1, 2, \cdots, N
\end{aligned}$$

with

$$g(U_G, U_D, U_s) = (\max(U_G - U_S - U_T, 0))^2 - (\max(U_G - U_D - U_T, 0))^2$$

The normalized parameters are: $R = 5 \cdot 10^3$, $C = 0.2 \cdot 10^{-3}$, $U_{op} = 5$, $U_T = 1$ and $K = 10^{-2}$. The output for a given input signal is shown in Figure 1.

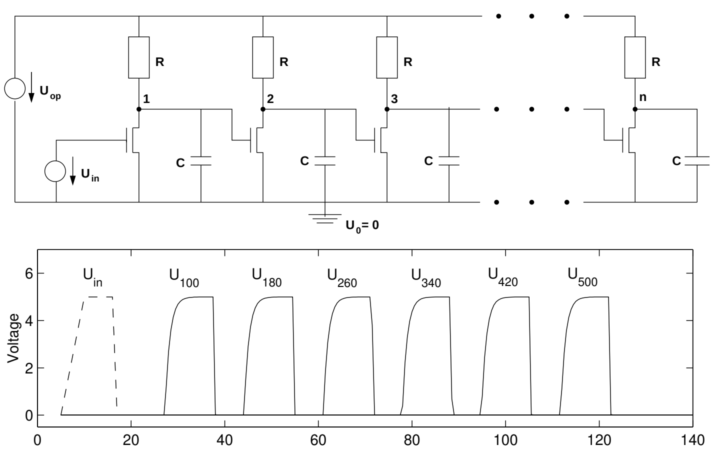

Fig. 1. The inverter chain

4.1 Technical details

The partitioning into active and latent components depends on the activity level of each inverter. The strategy is

a) y_i is active if $|y_i| > ActLevel$.
b) If y_i is active by a), so is $y_{i+1}, \ldots, y_{i+N_W}$.

The condition *b)* takes care of components that will turn active during the macro step. The size of the macro step is restricted by some H_{max}. In our experiments, $ActLevel = 0.7$, $N_W = 2$ and $H_{max} = 0.5$.

For the compound step, mixed multirate employs an instance of order (3)2, but for the later micro steps uses a method 3(2). That is to say, the thrid order solution is used to compute succeeding approximations in the later micro steps, only (cf. [1]).

MATLAB's built-in ROW-type scheme `ode23s` has been used as the underlying scheme in the operator-splitting method.

4.2 Results

The inverter chain problem, with 500 inverters were solved by the mixed multirate and the splitting method, and compared with `ode23s`. The results are given in Table 1.

From this table we observe a considerable gain in efficiency by using multirate methods. In terms of scalar function evaluations, the speedup are about 10 for both methods. In terms of floating point operations, the speedups are 2.5 and 4.6, respectively. This reduction in speedup is caused partly by the inevitable overhead of multirating, and partly by the fact that both multirate codes are experimental and far from optimised. We also note that `ode23s`

	Macro/micro steps	Rejected macro/ micro steps	Scalar function evaluations	flop
`ode23s`	5445	1625	$9.80 \cdot 10^6$	$5.1 \cdot 10^8$
Mixed multirate	287/11175	6/127	$0.93 \cdot 10^6$	$2.0 \cdot 10^8$
Operator splitting	260/5232	3/1439	$1.02 \cdot 10^6$	$1.1 \cdot 10^8$

Table 1. Test results for the stiff inverter chain problem.

and the operator splitting method use about the same number of micro steps, which makes sense since `ode23s` is used to integrate the active part over each macrostep. The mixed multirate method however, uses about twice as many micro steps, indicating that the step size selection scheme used here is somewhat more conservative than the scheme used in `ode23s`.

References

1. Bartel, A.: multirate ROW Methods of Mixed Type for Circuit Simulation. Submitted to SCEE-2000 Proceedings.
2. Günther, M. (1998): Simulating digital circuits numerically-a charge-oriented ROW approach. Numer. Math. 79, 203-212.
3. Günther, M., Hoschek, M., Weiner, R.: ROW methods adapted to cheap Jacobian. To appear in *Appl. Num. Math.*
4. Günther, M., Kværnø, A., Rentrop, P.: multirate Partitioned Runge-Kutta Methods. Submitted for publication in BIT.
5. Hairer, E. (1981): Order Conditions for Numerical Methods for Partitioned Ordinary Differential Equations. Numer. Math. **36**, pp. 431-445.
6. Kaps, P., Rentrop, P. (1979): Generalised Runge-Kutta methods of order four with step size control for stiff ordinary differential equations. Numer. Math. **30**, pp. 55-68.
7. Roche, M. (1988): Rosenbrock methods for differential algebraic equations. Numer. Math. **28**, pp. 145-162.
8. Strang, G. (1968): On the construction and comparison of difference schemes. SIAM J. Numer. Anal. **5**.
9. Yoshida, H. (1990): Construction of higher order symplectic integrators. Physics Letters A **150**.