Low Order Multirate Runge-Kutta Methods in Electric Circuit Simulation

A. Kværnø* and P. Rentrop†

February 12, 1999

Abstract

Coupling of different units from electric circuits, multibody dynamics and pneumatics lead to different time constants in dynamical simulation. Multirate Runge-Kutta methods, which use an inherent stepsize for each unit can possibly handle this situation in an efficient way. Order conditions and a coefficient set for an embedded low order explicit multirate scheme of order (2)3 are presented. The performance is tested for an inverter chain from electric circuits.

1 Introduction

In TCAD – technology computer aided design – numerical algorithms form the kernel for the industrial simulation packages. The algorithmic development and improvement is part of the success of Scientific Computing. During the last twenty years the simulation of subsystems was the main task, today coupled systems are in the focus. Their dynamical behaviour is characterized by different time constants, e.g. an analogue circuit runs in the microsecond scale, whereas the digital part is measured in nanoseconds. This requires integration schemes using the inherent stepsize of each subcircuit. Similar problems arise if different units from electric circuits, multibody dynamics and pneumatics are coupled.
In general the dynamics of the whole system can be described by an initial value problem of ODEs:

$$\begin{aligned} \dot{y}(t) &= f(t,y), \quad && y \in \mathbb{R}^n, \\ y(t_0) &= y_0, && t > t_0. \end{aligned} \tag{1}$$

Due to its technical background a partition of the system into subsystems is quite natural, so that

$$\begin{aligned} \dot{y}^{(1)} &= f_1(t, y^{(1)}, ..., y^{(p)}), \qquad && y^{(1)} \in \mathbb{R}^{n_1}, \\ &\vdots \\ \dot{y}^{(p)} &= f_p(t, y^{(1)}, ..., y^{(p)}), && y^{(p)} \in \mathbb{R}^{n_p}, \end{aligned}$$

*Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway

†Fachbereich 4 - Mathematik Technische Universität Darmstadt, Schloßgartenstr. 7, D-64289 Darmstadt, Germany

where

$$y = (y^{(1)}, ... y^{(p)})^T, \quad y(t_0) = y_0 \quad \text{and} \quad n = n_1 + ... + n_p.$$

For the numerical solution of the splitted system several approaches exists. Waveform relaxation schemes exploit well-known iteration techniques from the solution of large linear equation systems. We want to deal with multirate schemes, where different stepsizes are introduced explicitly in the discretization. Till now the mathematical literature on multirate schemes is limited. Andrus [1] describes a multirate algorithm based on a Runge-Kutta formula. Basic work for BDFs can be found in Andrus [2], Gear and Wells [6], Skelboe and Anderson [11], [12]. Multirate extrapolation schemes are successfully used in stellar problems by Engstler and Lubich [4], [5]. Their approach starts with a variable order method, where the fast components are integrated by the high order, for the major part of the slow components and for several integration steps a low order scheme is sufficient. In Günther and Rentrop [7], [8] a multirate ROW method is discussed, where the coupling is realized by interpolating and extrapolating state variables. Due to our knowledge a concise theory for multirate Runge-Kutta schemes is till now missing.
In section 2 we introduce some basics on multirate strategies and define our multirate Runge-Kutta approach. Section 3 includes the order conditions and a coefficient set of a order 3(2) scheme, which is developed around the Bogacki-Shampine pair [3], the base of a MATLAB program. In section 4 we discuss details of the implementation of the stepsize and multirate strategy. Section 5 includes a real-life example from electric circuits simulation. For an inverter chain the performance of the presented approach is discussed. The Appendix gives the necessary details on how order conditions are achieved.

2 Multirate Runge-Kutta Methods

In the following, the we assume that the solution $y(t)$ of the initial value problem (1) is split into active components $y_A(t)$, hopefully a very small subset of $y(t)$, and latent components $y_L(t)$, thus

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

The result is a split system,

$$\begin{aligned} \dot{y}_A(t) &= f_A(y_A, y_L), \qquad & y_A(t_0) &= y_{A,0}, \\ \dot{y}_L(t) &= f_L(y_A, y_L), \qquad & y_L(t_0) &= y_{L,0} \end{aligned} \tag{2}$$

which for simplicity, but without loss of generality, is assumed autonomous. The active components y_A are integrated with the small stepsize h, the latent components y_L are integrated with the large stepsize H. The methods used for the integration can, but do not have to, be the same. A synchronization of the *microsteps* h and the *macrosteps* H is performed at the end of each macrostep. The definition of a multirate Runge-Kutta method — MRK method — for the numerical solution of (2) becomes:

- For the macrostep H and the microstep h we have

$$h = (1/m) \cdot H. \tag{3.a}$$

- The active components y_A are defined by

$$k_{A,i}^{\lambda} = f_A(y_{A,\lambda} + h \sum_{j=1}^{s} a_{ij}\, k_{A,j}^{\lambda},\, \tilde{Y}_{L,i}^{\lambda}), \quad i = 1, 2, \ldots, s, \tag{3.b}$$

$$y_{A,\lambda+1} = y_{A,\lambda} + h \sum_{i=1}^{s} b_i\, k_{A,i}^{\lambda} \;\approx\; y(t_0 + (\lambda+1)h), \tag{3.c}$$

for $\lambda = 0, 1, 2, \ldots, m-1$.
Here, $c_i = \sum_{j=1}^{s} a_{ij}$ and $\tilde{Y}_{L,i}^{\lambda} \approx y_L(t_0 + (\lambda + c_i)h)$.

- The latent components y_L are defined by

$$k_{L,i} = f_L(\tilde{Y}_{A,i},\, y_{L,0} + H \sum_{j=1}^{\bar{s}} \bar{a}_{ij}\, k_{L,j}), \qquad j = 1, 2, \ldots, \bar{s}, \tag{3.d}$$

$$y_{L,1} = y_{L,0} + H \sum_{i=1}^{\bar{s}} \bar{b}_i\, k_{L,i} \;\approx\; y_L(t_0 + H), \tag{3.e}$$

where $\bar{c}_i = \sum_{j=1}^{\bar{s}} \bar{a}_{ij}$ and $\tilde{Y}_{A,i} \approx y_A(t_0 + \bar{c}_i\, H)$.

The coupling from the latent to the active subsystem and vice versa is performed by the intermediate stage values $\tilde{Y}_{A,i}$ and $\tilde{Y}_{L,i}^{\lambda}$. These can be computed by extrapolation or interpolation according to the following strategies:

Fastest first strategy:

- Integrate the active components y_A with m steps of stepsize h; y_L-values are obtained by extrapolation from the previous step.
- Integrate the latent components y_L with one macrostep H; y_A-values are obtained by interpolation.

Slowest first strategy:

- Integrate the latent components y_L with one macrostep H; y_A-values are obtained by extrapolation from the previous step.
- Integrate the active components y_A with m microsteps; y_L-values are interpolated.

A third option is to replace interpolation with extrapolation. Gear and Wells [6] use these strategies with multistep schemes, Günther and Rentrop [7], [8] with a Rosenbrock-Wanner method.

The extrapolation/interpolation strategy is a natural choice, but the approach inevitably turns the Runge-Kutta method into a two-step method. A genuine one(macro)step method can be achieved, with no extra cost, by the following formulas for the stage values:

$$\tilde{Y}_{L,i}^{\lambda} = y_{L,0} + h \sum_{j=1}^{\bar{s}} (\gamma_{ij} + \eta_j(\lambda))\, k_{L,j}, \quad i = 1, 2, \ldots s, \quad \lambda = 0, 1, \ldots, m-1, \tag{3.f}$$

$$\tilde{Y}_{A,i} = y_{A,0} + H \sum_{j=1}^{s} \bar{\gamma}_{ij}\, k_{A,j}^{0}, \quad i = 1, 2 \ldots, \bar{s}. \tag{3.g}$$

This resembles the slowest first strategy in the sense that y_A-values are obtained by extrapolation from the first microstep. A better idea might be to compute the $\tilde{Y}_{A,i}$-values by some low order method as explained in the next subsection. To distinguish the two methods, we refer to the one defined by (3.a-g) as MRKI, and the one with $\tilde{Y}_{i,A}$ computed with a low order method by MRKII.

Construction of a Third Order Explicit MRK-Method

Given two third order RK-methods (b_i, a_{ij}, c_i) and $(\bar{b}_i, \bar{a}_{ij}, \bar{c}_i)$ for the integration of the active and the latent components respectively. These methods are joined in an MRK by the coupling coefficients γ_{ij}, $\bar{\gamma}_{ij}$ and $\eta_j(\lambda)$, which must be chosen carefully to retain the order of the method. To do so, we need an order theory for the MRK method. In the appendix, the MRK is rewritten as a partitioned Runge-Kutta method. Applying the well established order theory for those (see e.g. [9, II.15]) the order conditions for MRK are found. The results for MRKI read as follows[1]:

If the simplifying assumptions

$$\sum_{j=1}^{\bar{s}} \eta_j(\lambda) = \lambda, \qquad \sum_{j=1}^{\bar{s}} \gamma_{ij} = c_i, \quad i = 1, 2, \dots, s \tag{4}$$

and

$$\sum_{j=1}^{s} \bar{\gamma}_{ij} = \bar{c}_i, \quad i = 1, 2, \dots, \bar{s} \tag{5}$$

are satisfied, then no extra conditions are required to ensure the order to be two. To be of order three the following conditions, in additions to the classical ones, have to be fulfilled:

$$\sum_{i=1}^{s} \sum_{j=1}^{\bar{s}} b_i \gamma_{ij} \bar{c}_j = 1/(6m), \qquad \sum_{i=1}^{s} \sum_{j=1}^{\bar{s}} b_i \eta_j(\lambda) \bar{c}_j = \lambda(\lambda+1)/(2m), \tag{6}$$

and

$$\sum_{i=1}^{\bar{s}} \sum_{j=1}^{s} \bar{b}_i \bar{\gamma}_{ij} c_j = m/6. \tag{7}$$

Note that what has so far been said about MRKI applies to explicit as well as implicit methods, and even to a mix of those, like implicit method for the latent part, and explicit for the active. In the following, we only consider explicit methods, thus a_{ij}, $\bar{a}_{ij}$, γ_{ij} and $\bar{\gamma}_{ij}$ are all 0 whenever $i \leq j$.

We have constructed an explicit third order MRKI method based on the Bogacki-Shampine (2)3 scheme [3], which also forms the base of a integration routine in MATLAB. The same method is used to integrate both sets of components. The Bogacki-Shampine order three method possesses three stages, however, one extra stage is added for error estimation. The error estimates are found by

$$\mathrm{err}_{A,\lambda+1} = h \sum_{i=1}^{4} d_i k_{A,i}^{\lambda}, \qquad \mathrm{err}_{L,1} = H \sum_{i=1}^{4} \bar{d}_i k_{L,i} \tag{8}$$

[1] Here, the assumptions (4) and (5) correspond to (19) in the appendix, (6) and (7) are given in Table 6.

$a_{21} = 1/2,$	$a_{31} = 0,$	$a_{32} = 3/4,$		
$b_1 = 2/9,$	$b_2 = 1/3,$	$b_3 = 4/9,$		
$d_1 = -5/72,$	$d_2 = 1/12,$	$d_3 = 1/9,$	$d_4 = -1/8,$	
$\gamma_{21} = 1/2,$	$\gamma_{31} = (3/4)\cdot(1-1/m),$	$\gamma_{32} = 3/(4m),$		
$\eta_1 = (-1/m + 3/2 - m/4)\cdot\lambda - 1/(2m)\cdot\lambda^2,$		$\eta_2 = (1/m - 3/2 + 3m/4)\cdot\lambda - 1/(2m)\cdot\lambda^2,$		
$\eta_3 = (1 - m/2)\cdot\lambda + (1/m)\cdot\lambda^2$	,			
$\bar{\gamma}_{21} = 1/2,$	$\bar{\gamma}_{31} = 3(1-m)/4,$	$\bar{\gamma}_{32} = 3m/4,$		only for MRKI
$\bar{\gamma}_{3,m/2} = -5/(4m),$	$\bar{\gamma}_{3,3m/4} = 13/(4m),$	$\bar{\gamma}_{i,\lambda} = 1/m$ otherwise,		only for MRKII
$\xi_{ij}(\lambda) = \lambda/i.$				

Table 1: Coefficients of MRK(2)3

where

$$k^{\lambda}_{A,4} = f_A(y_{A,\lambda+1}, y_{L,0} + h\sum_{j=1}^{3}(\eta_j(\lambda+1))k_{L,j}), \quad k_{L,4} = f_L(y_{A,m}, y_{L,1})$$

for the active and the latent components respectively. Thus, this extra stage requires no extra function evaluations.

One possible set of coefficients are given in Table 1. It should be remarked that the choice of coupling coefficients is not unique. However, our choice ensures that both sets of components are integrated with the same method if the size of micro- and macrosteps coincides ($m = 1$).

Computing $\tilde{Y}_{A,i}$ by an explicit Euler method might give a more stable method, but the order will be reduced to two. Fortunately, a slight modification is all that is needed to regain the order.

Again, let two third order explicit Runge-Kutta methods be given, and assume that $\bar{c}_i < \bar{c}_{i+1}$. Further, the number of microsteps m within the macrostep is chosen so that $m\bar{c}_i$ are integers. The following algorithm together with (3.a-f) forms the alternative MRK method called MRKII.

$$\begin{aligned}
&v_0 = y_{A,0}, \quad k_{L,1} = f_L(y_{A,0}, y_{L,0})\\
&\text{for } i = 1, 2, \ldots \bar{s}-1\\
&\quad \text{for } \lambda = \bar{c}_i m, \ldots, \bar{c}_{i+1}m - 1\\
&\qquad l_{\lambda+1} = f_A(v_\lambda, y_{L,0} + h\sum_{j=1}^{i}\xi_{ij}(\lambda)k_{L,j})\\
&\qquad v_{\lambda+1} = v_\lambda + h\cdot l_{\lambda+1}\\
&\quad \text{end}\\
&\quad \tilde{Y}_{A,i+1} = y_{A,0} + H\sum_{\lambda=1}^{\bar{c}_{i+1}m}\bar{\gamma}_{i+1,\lambda}l_\lambda \qquad (3.\text{g}')\\
&\quad k_{L,i+1} = f_L(\tilde{Y}_{A,i+1}, y_{L,0} + H\sum_{j=1}^{i}\bar{a}_{i+1,j}k_{L,j})\\
&\text{end}
\end{aligned}$$

Note that this algorithm assumes an explicit method for the latent components.

Now, the simplifying assumption (5) is replaced by[2]

$$\sum_{j=1}^{i} \xi_{ij}(\lambda) = \lambda, \qquad \sum_{\lambda=1}^{\bar{c}_i m} \bar{\gamma}_{i,\lambda} = \bar{c}_i, \quad i = 1, 2, \ldots, \bar{s}, \tag{9}$$

and the order three condition (7) by

$$\sum_{i=2}^{\bar{s}} \sum_{\lambda=1}^{\bar{c}_i m} \bar{b}_i \bar{\gamma}_{i\lambda}(\lambda - 1) = m/6. \tag{10}$$

The coefficients γ_{ij} and $\eta_j(\lambda)$ are not altered by these changes. In the computation of l_λ, we can use the average of all available $k_{L,i}$-values to find a rough approximation to $y_L(t_0 + \lambda h)$. Thus, $\xi_{ij}(\lambda) = \lambda/i$ and the first condition of (9) is satisfied. If we decide to use the forward Euler solution directly, that is $\tilde{Y}_{A,i} = v_{\bar{c}_i m}$, it means that $\bar{\gamma}_{ij} = 1/m$. This is perfectly fine what the second condition of (9) concerns, but the left hand side of (10) becomes

$$\sum_{i=2}^{\bar{s}} \bar{b}_i \sum_{\lambda=1}^{\bar{c}_i m} (\lambda - 1)/m = \sum_{i=1}^{\bar{s}} (m\bar{b}_i\bar{c}_i^2 - \bar{b}_i\bar{c}_i)/2 = m/6 - 1/4$$

since $\bar{c}_1 = 0$, $\sum \bar{b}_i\bar{c}_i^2 = 1/3$ and $\sum \bar{b}_i\bar{c}_i = 1/2$. Thus, the order is reduced to 2. To change the last of the stage values, so that $\tilde{Y}_{A,\bar{s}} = v_{\bar{c}_{\bar{s}}m} + h\alpha(l_{\bar{c}_{\bar{s}}m} - l_{\bar{c}_{\bar{s}-1}m})$ is one possible remedy. In that case, $\bar{\gamma}_{\bar{s},\bar{c}_{\bar{s}-1}m} = (1-\alpha)/m$ and $\bar{\gamma}_{\bar{s},\bar{c}_{\bar{s}}m} = (1+\alpha)/m$, all the others $\bar{\gamma}_{i,\lambda}$ are still $1/m$. Thus, the left hand side of (10) becomes

$$\sum_{i=2}^{\bar{s}} \sum_{\lambda=1}^{\bar{c}_i m} \bar{b}_i \bar{\gamma}_{i\lambda}(\lambda - 1) = m/6 - 1/4 + \alpha\bar{b}_{\bar{s}}(\bar{c}_{\bar{s}} - \bar{c}_{\bar{s}-1})$$

and the condition is satisfied if $\alpha = (4\bar{b}_{\bar{s}}(\bar{c}_{\bar{s}} - \bar{c}_{\bar{s}-1}))^{-1}$. In the case of the Bogacki-Shampine scheme, that gives $\alpha = 9/4$.

This MRKII method with coefficients given in Table 1, is implemented in the code MRK(2)3 as explained in the next section.

3 Technical Details of Implementation

Selection of new stepsizes and partitioning into latent and active components are performed at the end of each macrostep. The strategies are similar to those proposed by Günther and Rentrop [7], [8], but modified to suit the needs of an explicit method. The stepsize selection goes as follows: Based on the error estimate (8) a new stepsize h_{new} is proposed for each component. In addition, a stiffness check is carried out, by proposing an alternative stepsize based on an error estimate with low order, but with a large stability region as explained by Hairer et.al. [10, IV.2]. If the alternative stepsize is larger than h_{new}, the component is marked stiff. The new macrostepsize H_{new} is

$$H_{new} = \min(\min_{stiff}(h_{new}), 0.5 \cdot \max_{all}(h_{new})).$$

Further, H is not allowed to increase nor to decrease too fast, thus H_{new} is limited by

$$0.5 \cdot H \leq H_{new} \leq 1.5 \cdot H.$$

[2]The assumptions (9) correspond to (20) in the appendix, and the condition (10) to (21).

For the microstepsize we choose $\min_{all}(h_{new})$, adjusted so that the number of microsteps within the macrostep is divisible by four, to ensure that $c_i m$ are integers.

The individual stepsize proposal h_{new} also serves as a partitioning criterion: If $h_{new} \geq H_{new}$ then the component is treated as latent, otherwise as active. Thus, all stiff components are treated as latent. Additionally, including information on the neighbourhood of active components in the partitioning strategy minimizes its rate of rejection. The inverter chain can serve as an example: The runtime of a signal is delayed by passing through the circuit, i.e. a window of active elements passes through the circuit in time. A partitioning strategy can exploit this information by generating a window of active components $[n_{min}, n_{max}]$ with n_{min} or n_{max}, respectively the minimum and the maximum number, respectively, of an active element. To increase the reliability of the partitioning strategy, the neighbors of active components could be activated, i.e. the window would be enlarged. In other respects technical information can be exploited. To avoid a leap over an active phase, the maximum macro stepsize has to be limited by the length of active phases.

4 Numerical Simulation Results

4.1 Mathematical Model of an Inverter Chain

A MOSFET – **m**etal **o**xide **s**emiconductor **f**ield **e**ffect **t**ransistor – inverter, see Figure 1 transforms a given digital signal from the logical value 0 to 1 and vice versa. The logical value 1 stands for a constant operating voltage U_{op}, the logical value 0 is the ground voltage U_0. The voltage U_T characterizes a threshold, such that any input signal U_{in} greater than $U_{op} - U_T$ gives the logical value 1, otherwise 0.

The mathematical model of a circuit is based on Kirchhoff's laws:

KCL1: The sum of currents traversing each cutset of the network is 0.

KCL2: The sum of voltages around each loop of the network is 0.

The main modelling principles are:

- There are five basic elements: resistor, capacitor, inductor (in analogue circuits), voltage and current sources.
- The (non)linear controlled voltage or current source and the resistor describe the static behaviour.
- the (non)linear controlled capacitors or inductors describe the dynamic behaviour.

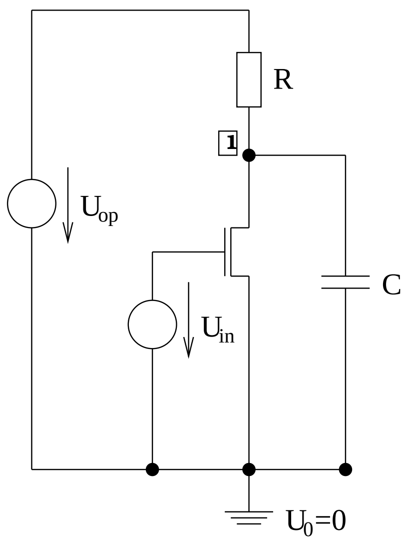

Figure 1: Circuit of a single MOSFET inverter

The single inverter, see Figure 1, consists of voltage sources, one capacitor, one resistor and one MOSFET. The mathematical model for the MOSFET is derived from a subcircuit having an appropriate voltage-current characteristic. From the hierarchy of models we choose the simplest one, a voltage controlled current source. The current source fits the drain current with respect to the gate voltage. Its characteristics are:

$$
\begin{aligned}
&I_{GS} = I_{GD} = I_{GB} = 0, \qquad &&\text{B: bulk, D: drain, G: gate, S: source,}\\
&I_{BD} = I_{BS} = 0, &&\\
&I_{DS} = K \cdot g(U_G, U_D, U_S), &&
\end{aligned}
$$

where g is defined piecewise:

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

The technical parameters are: $K = 2 \cdot 10^{-4}$ [A V^{-2}], $U_T = 1$ [V], $U_{op} = 5$ [V].

The modified nodal voltage approach applied to node 1 in the inverter circuit leads to node 1: $I_{resistor} + I_{capacitor} + I_{MOSFET} = 0$
Applying Ohms law, assuming a constant capacitor and using $U_0 = 0$ gives

$$\dot{U}_1 = (U_{op} - U_1)/(RC) - K \cdot g(U_{in}, U_1, U_0)/C \tag{11}$$

Since the run time of a digital circuit is measured in nanoseconds, we scale by a factor 10^9. Choosing realistic values for the resistor $R = 5 \cdot 10^3$ and for the capacitor $C = 0.2 \cdot 10^{-12}$ we end up with the normalized equation

$$\dot{U}_1 = (U_{op} - U_1) - g(U_{in}, U_1, U_0) \tag{12}$$

The output of an inverter can be fed as the gate-input of a next inverter and so on. This creates an inverter chain. Charging and discharging of the capacitors produces delays in the run time of the signal. For an even number n_i of inverters one expects that the last output signal is equal to a phase shifted input. The normalized mathematical model for an inverter chain is straightforward:

$$
\begin{aligned}
\dot{U}_1 &= (U_{op} - U_1) - g(U_{in}, U_1, U_0)\\
\dot{U}_k &= (U_{op} - U_k) - g(U_{k-1}, U_k, U_0) \quad \text{for } k = 2, ..., n_i.
\end{aligned}
$$

Remark: In electric devices an inverter chain is used to achieve a well-defined time delay of signals. If the output of the last inverter is fed back as the input of the first inverter, the circuit acts as a ring oscillator.

The inverter chain is well suited as a test example. The solution structure is very regular, so one can easily check the precision of the numerical simulation. The size of the circuit can be easily enlarged. The circuit contains a small time dependent active part of 20 to 40 inverters depending on the prescribed input signal, and a large latent part.
As input signal U_{in} was chosen the polygon with edges

$$U_{in}(0) = 0, \quad U_{in}(5) = 0, \quad U_{in}(10) = 5, \quad U_{in}(15) = 5, \quad U_{in}(17) = 0 \quad \text{and} \quad U_{in}(t > 17) = 0.$$

The length of this input signal requires approximately 20 active inverters.

Number of inverters	50	100	200	400	800
NSTEP	509	844	1515	2864	5573
NSTEPREJ	5	5	5	5	5
ERROR	4.e-5	5.e-5	7.e-5	8.e-5	4.e-5
TIME	0.72	2.5	8.7	33	137

Table 2: Test results on RK(2)3 for the inverter chain

Number of inverters	50	100	200	400	800
NSTEPMAK	206	295	473	829	1541
NSTEPMAKREJ	7	7	7	7	7
NSTEPMIK	609	953	1661	3085	5933
NSTEPMIKREJ	1	1	1	1	1
ERROR	3.e-5	3.e-5	3.e-5	1.e-5	7.e-6
TIME	0.35	1.0	3.4	13	52

Table 3: Test results on MRK(2)3 for the inverter chain

4.2 Numerical Results

The proposed method MRK(2)3 was tested for an inverter chain with 50, 100, 200, 400 and 800 inverters in comparison with the same code with the mulitrating turned off (abbreviated to RK(2)3), and the A-stable multirate Rosenbrock-Wanner method MROW(2)3 by Günther and Rentrop [7]. For all methods, the initial stepsize HI=1.e-2 was used, and the prescribed tolerance was TOL=1.e-3. According to the discontinuities of the input signal U_{in}, the integration was restarted at $t = 5, 10, 15$ and 17. The maximum macrostepsize was 1.145. The linear equations system of MROW(2)3 was solved by full Gaussian elimination techniques. The test results are listed in Tables 2, 3 and 4.

The following abbreviations are used:

NSTEP is the number of steps (only for RK(2)3).
NSTEPREJ is the number of rejected steps (only for RK(2)3).
NSTEPMAK is the number of macrostep (for MRK(2)3 and MROW(2)3).
NSTEPMAKREJ is the number of rejected macrosteps (for MRK(2)3 and MROW(2)3).
NSTEPMIK is the number of microsteps (for MRK(2)3 and MROW(2)3).
NSTEPMIKREJ is the number of rejected microsteps (for MRK(2)3 and MROW(2)3).
ERROR is the maximum error of solution components at the end of the interval.
TIME is the total computing time in seconds to solve the problem. Computations were performed in FORTRAN double precision on an INDY R5000.

The statistics of the MRK(2)3 scheme show that the stepsize control works very reliable for both the active and the latent components. Also the partitioning strategy is very reliable. With exception of some oscillations in the beginning of the integration, the number of active components stays between 20 and 25. Especially compared to the MROW(2)3 scheme with its interpolation/extrapolation strategy the number of rejected microsteps are negligible for the new approach. The better performance of MRK(2)3 in terms of computation time is less

Number of inverters	50	100	200	400	800
NSTEPMAK	99	124	175	275	474
NSTEPMAKREJ	5	5	5	5	6
NSTEPMIK	320	571	1074	2074	4068
NSTEPMIKREJ	81	149	265	543	1084
ERROR	3.e-7	3.e-7	3.e-7	4.e-7	1.e-7
TIME	1.5	7.0	36	221	2117

Table 4: Test results on MROW(2)3 for the inverter chain

surprising, since MROW(2)3 as a stiff integrator has to solve linear equation systems. The standard linear equation solver in MROW(2)3 is not optimized for the inverter chain.

4.3 Conclusion

The first numerical experiments with a genuine multirate Runge-Kutta approach are very promising. The explicit method with stepsize control and partitioning strategy has worked reliable in the inverter chain. To our opinion technical problems with different time constants cannot be seen as standard tasks. At least today one only can expect that numerical methods must be tailored to a given application. In this sense the presented approach offers new possibilities for the numerical simulation of technical problems.

5 Acknowledgement

P. Rentrop thanks the Department of Mathematics at NTNU Trondheim for their great hospitality and is indebted to the SYNODE project for financial support. The SYNODE project is sponsored by the Norwegian Research Council under contract no. 111038/410.

A Order Conditions for Multirate Runge-Kutta Methods

There are several ways to derive order conditions for MRK-methods. The most appropriate might be to derive a Butcher-series type theory for the methods. In this paper we only aim for methods of order up to three, thus it seems more convenient to take advantage of the already established theory of partitioned Runge-Kutta methods (PRK), described by Hairer et al. [9, II.15].

Partitioned Runge-Kutta Methods

A partitioned Runge-Kutta method (PRK) applied to the split system (2) is given by

$$k_{A,i} = f_A(y_{A,0} + H\sum_{j=1}^{q} \alpha_{ij}\, k_{A,j}, y_{L,0} + H\sum_{j=1}^{r} \beta_{ij}\, k_{L,j}), \quad i = 1,2,\dots,q, \tag{13.a}$$

$$y_{A,1}^H = y_{A,0} + H\sum_{i=1}^{q} b_{A,i}\, k_{A,i} \approx y_A(t_0+H), \tag{13.b}$$

$$k_{L,i} = f_L(y_{A,0} + H\sum_{j=1}^{q} \xi_{ij}\, k_{A,j}, y_{L,0} + H\sum_{j=1}^{r} \delta_{ij}\, k_{L,j}), \quad i = 1,2,\dots,r, \tag{13.c}$$

$$y_{L,1}^H = y_{L,0} + H\sum_{i=1}^{r} b_{D,i}\, k_{L,i} \approx y_L(t_0+H). \tag{13.d}$$

The coefficients can be presented in an extended Butcher-tableau like

$$\begin{array}{c|ccc|ccc|c}
c_{A,1} & \alpha_{11} & \cdots & \alpha_{1q} & \beta_{11} & \cdots & \beta_{1r} & c_{B,1} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
c_{A,q} & \alpha_{q1} & \cdots & \alpha_{qq} & \beta_{q1} & \cdots & \beta_{qr} & c_{B,q} \\
\hline
c_{C,1} & \xi_{11} & \cdots & \xi_{1q} & \delta_{11} & \cdots & \delta_{1r} & c_{D,1} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
c_{C,r} & \xi_{r1} & \cdots & \xi_{rq} & \delta_{r1} & \cdots & \delta_{rr} & c_{D,r} \\
\hline
 & b_{A,1} & \cdots & b_{A,q} & b_{D,1} & \cdots & b_{D,r} &
\end{array}
\quad \text{or shorter} \quad
\begin{array}{c|c|c|c}
\hat{c}_A & \hat{A} & \hat{B} & \hat{c}_B \\
\hat{c}_C & \hat{C} & \hat{D} & \hat{c}_D \\
\hline
 & \hat{b}_A^T & \hat{b}_D^T &
\end{array}. \tag{14}$$

Here, $c_{A,i} = \sum_{j=1}^{q} \alpha_{ij}$, $c_{B,i} = \sum_{j=1}^{r} \beta_{ij}$ and so on. Using the vector notation, the method (13) can be rewritten as

$$\mathbf{k}_A = \mathbf{f}_A(\mathbb{1}_q \cdot y_{A,0} + H \cdot \hat{A}\,\mathbf{k}_A,\, \mathbb{1}_q \cdot y_{L,0} + H \cdot \hat{B}\,\mathbf{k}_L), \tag{15.a}$$

$$y_{A,1}^H = y_{A,0} + H \cdot \hat{b}_A^T\,\mathbf{k}_A, \tag{15.b}$$

$$\mathbf{k}_L = \mathbf{f}_L(\mathbb{1}_r \cdot y_{A,0} + H \cdot \hat{C}\,\mathbf{k}_A,\, \mathbb{1}_r \cdot y_{L,0} + H \cdot \hat{D}\,\mathbf{k}_L), \tag{15.c}$$

$$y_{L,1}^H = y_{L,0} + H \cdot \hat{b}_D^T\,\mathbf{k}_D \tag{15.d}$$

with $\mathbb{1}_q = [1,1,\dots,1]^T \in \mathbb{R}^q$, similar for $\mathbb{1}_r$. For simplicity, the obvious tensor notation has been omitted. Actually, our definition coincide with the definition in [9] only if $\hat{C} = \hat{A}$ and $\hat{B} = \hat{D}$. However, the extension to our definition is straightforward, and results in the following:

If the simplifying assumptions

$$\hat{c}_B = \hat{c}_A, \qquad \hat{c}_C = \hat{c}_D \tag{16}$$

are satisfied, then the order conditions up to order 3 are given in Table 5. The right column gives the order conditions if an interpolant is used for the active components. These conditions will be used when analyzing the order of the microsteps for the active components. In the table, $\hat{b}_A(\theta)$ is a vector function of θ, so that $\hat{b}_A(0) = 0_s^T = [0,0,\dots,0]$ and $\hat{b}_A(1) = \hat{b}_A$, and θ takes values between zero and one.

If the methods $(\hat{b}_A, \hat{A}, \hat{c}_A)$ and $(\hat{b}_D, \hat{D}, \hat{c}_D)$ are both of (classical) order 3, then the only additional conditions are those given by v).

No.	Order	active		latent		interpolant, active	
i)	1	$\hat{b}_A^T \mathbb{1}_q$	$= 1$	$\hat{b}_D^T \mathbb{1}_r$	$= 1$	$\hat{b}_A^T(\theta) \mathbb{1}_q$	$= \theta$
ii)	2	$\hat{b}_A^T \hat{c}_A$	$= 1/2$	$\hat{b}_D^T \hat{c}_D$	$= 1/2$	$\hat{b}_A^T(\theta) \hat{c}_A$	$= \theta^2/2$
iii)	3	$\hat{b}_A^T \hat{c}_A^2$	$= 1/3$	$\hat{b}_D^T \hat{c}_D^2$	$= 1/3$	$\hat{b}_A^T(\theta) \hat{c}_A^2$	$= \theta^3/3$
iv)		$\hat{b}_A^T \hat{A} \hat{c}_A$	$= 1/6$	$\hat{b}_D^T \hat{D} \hat{c}_D$	$= 1/6$	$\hat{b}_A^T(\theta) \hat{A} \hat{c}_A$	$= \theta^3/6$
v)		$\hat{b}_A^T \hat{B} \hat{c}_D$	$= 1/6$	$\hat{b}_D^T \hat{C} \hat{c}_A$	$= 1/6$	$\hat{b}_A^T(\theta) \hat{B} \hat{c}_D$	$= \theta^3/6$

Table 5: Order conditions for PRK methods, presupposed $\hat{c}_B = \hat{c}_A$ and $\hat{c}_C = \hat{c}_D$

Order Conditions for MRKI.

Let $A = (a_{ij}) \in \mathbb{R}^{s \times s}$, $\bar{A} = (\bar{a}_{ij}) \in \mathbb{R}^{\bar{s} \times \bar{s}}$, $B(\lambda) = G + F(\lambda) = (\gamma_{ij} + \eta_j(\lambda)) \in \mathbb{R}^{s \times \bar{s}}$, $\bar{G} = (\bar{\gamma}_{ij}) \in \mathbb{R}^{\bar{s} \times s}$, $b = (b_i) \in \mathbb{R}^s$ and $\bar{b} = (\bar{b}_i) \in \mathbb{R}^{\bar{s}}$. The MRKI-method (3.b-e) can be written in vector form by

$$\mathbf{k}_A^\lambda = \mathbf{f}_A(\mathbb{1}_s \cdot y_{A,\lambda} + (H/m) \cdot A\, \mathbf{k}_A^\lambda,\ \mathbb{1}_s \cdot y_{L,0} + (H/m) \cdot B(\lambda)\, \mathbf{k}_L) \tag{17.b}$$

$$= \mathbf{f}_A(\mathbb{1}_s \cdot y_{A,0} + (H/m) \cdot \sum_{j=0}^{\lambda-1} \mathbb{1}_s \cdot (b^T \mathbf{k}_A^j) + (H/m) \cdot A\, \mathbf{k}_A^\lambda,\ \mathbb{1}_s \cdot y_{L,0} + (H/m) \cdot B(\lambda) \mathbf{k}_L),$$

$$y_{A,\lambda+1} = y_{A,\lambda} + (H/m) \cdot b^T \mathbf{k}_A^\lambda \ = \ y_{A,0} + (H/m) \cdot \sum_{j=0}^{\lambda} b^T \mathbf{k}_A^j, \tag{17.c}$$

$$\mathbf{k}_L = \mathbf{f}_L(\mathbb{1}_{\bar{s}} \cdot y_{A,0} + H \cdot \bar{G}\, \mathbf{k}_A^0,\ \mathbb{1}_{\bar{s}} \cdot y_{L,0} + H \cdot \bar{A}\, \mathbf{k}_L), \tag{17.d}$$

$$y_{L,1} = y_{L,0} + H \cdot \bar{b}^T \mathbf{k}_L. \tag{17.e}$$

Using $\mathbf{k}_A = [(\mathbf{k}_A^0)^T, (\mathbf{k}_A^1)^T, \dots, (\mathbf{k}_A^{m-1})^T]^T$ the MRK can be written as a PRK with extended tableau (14) like

$$\begin{array}{c|cccc|c|c}
\frac{1}{m}c & \frac{1}{m}A & & & & \frac{1}{m}B(0) & \frac{1}{m}c_{B,0} \\
\frac{1}{m}(\mathbb{1}_s + c) & \frac{1}{m}\mathbb{1}_s b^T & \frac{1}{m}A & & & \frac{1}{m}B(1) & \frac{1}{m}c_{B,1} \\
\vdots & \vdots & \vdots & \ddots & & \vdots & \vdots \\
\frac{1}{m}((m-1)\mathbb{1}_s + c) & \frac{1}{m}\mathbb{1}_s b^T & \frac{1}{m}\mathbb{1}_s b^T & \cdots & \frac{1}{m}A & \frac{1}{m}B(m-1) & \frac{1}{m}c_{B,m-1} \\
\hline
\bar{c}_G & \bar{G} & 0 & \cdots & 0 & \bar{A} & \bar{c} \\
\hline
 & \frac{1}{m}b^T & \frac{1}{m}b^T & \cdots & \frac{1}{m}b^T & \bar{b}^T &
\end{array} \tag{18}$$

assuming $b^T \mathbb{1}_s = 1$. Here $c_{B,\lambda} = B(\lambda)\mathbb{1}_{\bar{s}}$ and $\bar{c}_G = \bar{G}\mathbb{1}_s$. The simplifying assumptions (16) is satisfied if

$$F(\lambda)\, \mathbb{1}_{\bar{s}} = \lambda \mathbb{1}_s, \qquad G\, \mathbb{1}_{\bar{s}} = c, \quad \text{and} \quad \bar{G}\, \mathbb{1}_s = \bar{c}. \tag{19}$$

Table 6 lists the order conditions for the MRKI. The conditions corresponding to the latent

Order	active			latent		
1	$b^T\,\mathbb{1}_s$	$=$	1	$\bar b^T\,\mathbb{1}_{\bar s}$	$=$	1
2	$b^T\,c$	$=$	$1/2$	$\bar b^T\bar c^T$	$=$	$1/2$
3	$b^T\,c^2$	$=$	$1/3$	$\bar b^T\,\bar c^2$	$=$	$1/3$
	$b^T\,A\,c$	$=$	$1/6$	$\bar b^T\,\bar A\,\bar c$	$=$	$1/6$
	$b^T\,G\,\bar c$	$=$	$1/(6m)$	$\bar b^T\,\bar G\,c$	$=$	$m/6$
	$b^T\,F(\lambda)\,\bar c$	$=$	$\lambda(\lambda+1)/(2m)$			

Table 6: Order conditions for MRKI, presupposed (19).

components follows directly from Table 5 by comparing the tableau of the PRK (14) with that of the MRKI (18).

For the active components, the method should provide a solution of a certain order not only at the end of the macrostep, but also in the microsteps. To attend this problem, we observe that

$$y_{A,\lambda} = y_{A,0} + H\cdot \hat b_A^T\,(\lambda/m)\mathbf{k}_A \quad\text{with}\quad \hat b_A^T(\lambda/m) = (1/m)\cdot[\underbrace{b^T,b^T,\cdots,b^T}_{\lambda\text{ times}},\underbrace{0_s^T,0_s^T,\cdots,0_s^T}_{m-\lambda\text{ times}}].$$

Thus, the interpolant conditions of Table 5 apply, with $\theta = \lambda/m$, $\lambda = 1,2,\dots,m$. Since a limited number (that is λ) of steps does not change the order of the solution for an ordinary differential equations, the first four MRK-conditions imply the first four PRK-conditions[3]. The last PRK-condition v) becomes

$$\hat b_A(\lambda/m)\,\hat B\hat c_D = (1/m^2)\sum_{i=0}^{\lambda-1} b^T(G+F(i))\bar c = \lambda^3/(6m^3), \qquad \lambda = 1,2,\dots,m.$$

Since $F(0)=0$ we obtain $b^T\,G\,\bar c = 1/(6m)$. From this follows

$$\sum_{i=0}^{\lambda-1} b^T F(i)\bar c = (\lambda^3-\lambda)/(6m)$$

thus

$$b^T F(\lambda)\bar c = ((\lambda+1)^3-(\lambda+1))/(6m) - \sum_{i=0}^{\lambda-1} b^T F(i)\bar c = \lambda(\lambda+1)/(2m)$$

and the last MRK-condition for the active components is justified.

[3]This result can also be accomplished by a technical proof. As an example,

$$\hat b_A^T(\lambda/m)\hat c_A^2 = (1/m^3)\sum_{i=0}^{\lambda-1} b^T\,(i\mathbb{1}_s + c)^2 = (b^T\mathbb{1}_s)/m^3\sum_{i=1}^{\lambda-1} i^2 + (2b^T c)/m^3\sum_{i=1}^{\lambda-1} i + \lambda(b^T c^2)/m^3$$

which, if the classical order 3 conditions are satisfied, becomes $(1/3)\;(\lambda/m)^3$ as expected.

Order Conditions for MRKII

The MRKI method is changed to an MRKII method by adding m stages (of which only the first $\bar{c}_s m$ are actually computed) to the method for the active components. These are given by

$$\mathbf{l} = \mathbf{f}_A(\mathbb{1}_m \cdot y_{A,0} + H \cdot A_1\,\mathbf{l}, \mathbb{1}_m \cdot y_{L,0} + H \cdot B_1\,\mathbf{k}_L)$$

where A_1 is a strictly lower triangular $m \times m$ matrix whose elements are $1/m$ and B_1 is a $m \times \bar{s}$ matrix with elements ξ_{ij}/m, $\xi_{ij} = 0$ whenever $i < \bar{c}_j m$. The vector $\mathbf{l}$ are used in in the computation of $\mathbf{k}_L$, which becomes

$$\mathbf{k}_L = \mathbf{f}_L(\mathbb{1}_{\bar{s}} \cdot y_{A,0} + H \cdot \bar{G}\,\mathbf{l}, \mathbb{1}_{\bar{s}} \cdot y_{L,0} + H \cdot \bar{A}\,\mathbf{k}_L)$$

where $\bar{G}$ is an $\bar{s} \times m$ matrix with elements $\bar{\gamma}_{i\lambda}$, with $\bar{\gamma}_{i\lambda} = 0$ whenever $\bar{c}_i m < \lambda$. Thus, the PRK tableau for the MRKII is given by

$$\begin{array}{c|cc|c|c} c_{A_1} & A_1 & 0 & B_1 & c_{B_1} \\ \hat{c}_A & 0 & \hat{A} & \hat{B} & \hat{c}_B \\ \hline \hat{c}_C & \bar{G} & 0 & \hat{D} & \hat{c}_D \\ \hline & 0 & \hat{b}_A^T & \hat{b}_D^T & \end{array}.$$

Here $\hat{A}$, $\hat{B}$, $\hat{D}$, $\hat{b}_A$ and $\hat{b}_D$ are adopted from the MRKI.

Obviously, from the definition of A_1, $c_{A_1} = A_1 \mathbb{1}_m = (1/m) \cdot [0, 1, 2, \ldots, m-1]^T$. Thus, the assumptions (16) are given by the first two conditions of (19) and

$$B_1\,\mathbb{1}_{\bar{s}} = (1/m) \cdot [0, 1, 2, \ldots, m-1]^T \quad \text{and} \quad \bar{G}\,\mathbb{1}_m = \bar{c}. \tag{20}$$

If this is so, the only order condition from Table 5 affected by the modification is the last one for the latent components, which now becomes

$$\bar{b}^T\,\bar{G}\,[0, 1, \ldots, m-1]^T = m/6. \tag{21}$$

Thus, a set of order 3 conditions for MRKII are established.

References

[1] J. F. Andrus. Numerical solution of systems of ordinary differential equations separated into subsystems. *SIAM J. Numer Anal.*, 16:605–611, 1979.

[2] J. F. Andrus. Stability of a multirate method for numerical integration of ODEs. *Comp. Math. Appl.*, 25:3–14, 1993.

[3] P. Bogacki and L. F. Shampine. A 3(2) pair of Runge-Kutta formulas. *Appl. Math. Letters*, 2:1–9, 1989.

[4] C. Engstler and C. Lubich. Multirate extrapolation for differential equations with different time scales. *Computing*, 58:173–185, 1997.

[5] C. Engstler, C. Lubich. MUR8 – a multirate extension of the eigth order Dormand-Prince method. *APNUM*, 25:185 – 192, 1997

[6] C. W. Gear and D. R. Wells. Multirate linear multistep methods. *BIT*, 24:484–502, 1984.

[7] M. Günther and P. Rentrop. Multirate ROW methods and latency of electric circuits. *Appl. Num. Math.*, 13:83–102, 1993.

[8] M. Günther and P. Rentrop. Partitioning and multirate strategies in latent electric circuits. *International series of Numerical Mathematics*, 117:33–60, 1994.

[9] E. Hairer, S.P Nørsett, and G. Wanner. *Solving ordinary differential equations I, Nonstiff problems.* Springer-Verlag, second edition, 1993.

[10] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems.* Springer, second edition, 1996.

[11] S. Skelboe. Stability properties of backward differentiation multirate formulas. *Appl. Numer. Math.*, 5:151–160, 1989.

[12] S. Skelboe and P. U. Anderson. Stability properties of backward Euler multirate formulas. *SIAM J. Sci. Stat. Comp.*, 10:1000–1009, 1989