

Envelope Transient Simulation of Nonlinear Electronic Circuits Using Multi-Rate Runge-Kutta Algorithms

JORGE OLIVEIRA, ADÉRITO ARAÚJO*

Department of Electrical Engineering
Technology and Management School - Polytechnic Institute of Leiria
Morro do Lena, Alto Vieiro, Apartado 4163, 2411-901 Leiria

*Department of Mathematics

University of Coimbra
Apartado 3008, 3001-454 Coimbra
PORTUGAL

oliveira@estg.ipleiria.pt, alma@mat.uc.pt

Abstract: - Time-step integration is a popular technique commonly used for the envelope transient simulation of an electronic circuit. However, many kinds of circuits are characterized by widely separated time scales, which lead to significant computational costs when numerically solving its differential systems. Even so, this situation can be exploited in an efficient way using multi-rate methods, which integrate system components with different step sizes. In this paper two multi-rate Runge-Kutta schemes are studied and tested in terms of computational speed and numerical stability. The results for linear stability analysis here obtained are much more coherent with the characteristics of the methods than the ones previously presented in [6].

Key-Words: - Electronic circuit simulation, transient analysis, multi-rate Runge-Kutta methods, speed, stability.

1 Introduction

Transient analysis of an electronic circuit is usually expressed by an initial value problem of the form

$$y'(t) = f(y), \quad y(t_0) = y_0, \quad y \in \mathbb{R}^n, \quad t > t_0 \quad (1)$$

where $y(t)$ is the solution. This system of ordinary differential equations (ODEs) can be numerically solved by time-step integration, which is a classical technique that is commercially used by all SPICE like simulators. However, when integrating systems whose components evolve at different time scales one would like to use numerical methods that do not expend unnecessary work on slowly changing components. In such cases traditional time-step integrators become inefficient and numerical schemes with different time-step sizes are required. For example, in highly integrated electronic circuits normally only a small part of the elements is active, whereas the major part is latent. This latency can be exploited by multi-rate methods, which integrate components of the slow subsystem with a larger step length than the fast subsystem.

Let us consider (1). If we split this ordinary differential system into active and latent subsystems we obtain

$$\begin{aligned} y_A'(t) &= f_A(y_A, y_L), & y_A(t_0) &= y_{A,0} \\ y_L'(t) &= f_L(y_A, y_L), & y_L(t_0) &= y_{L,0} \end{aligned} \quad (2)$$

with

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

where y_A is the active components vector and y_L the latent components vector. The efficiency of the methods presented in this paper is verified only if there is a small number of fast changing components, i.e., if y_A is a small subset of y . It is so because while the active components y_A are integrated with a small step size h (microstep), the latent components y_L are integrated with a large step size H (macrostep). The number of microsteps within a macrostep is m , thus

$$h = (1/m) \cdot H, \quad m \in \mathbb{N}.$$

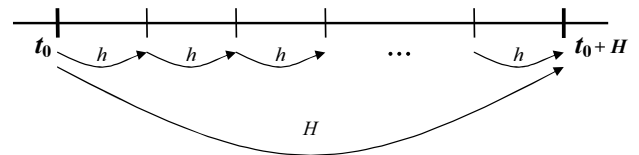


Fig.1. Macro and microsteps

Throughout the integration process the partition into fast and slow components may vary with time, as well as m . For a better understanding let us consider the basic example illustrated in Fig.2, where we have only five variables depending on t . Defining A as the active subsystem (fast changing components) and L

as the latent subsystem (slow changing components), then we will have

$$A = \{y_1, y_4\}, \quad L = \{y_2, y_3, y_5\}$$

for t in the interval $[t_0, t_1]$ and

$$A = \{y_2, y_5\}, \quad L = \{y_1, y_3, y_4\}$$

for $t \in [t_1, t_2]$.

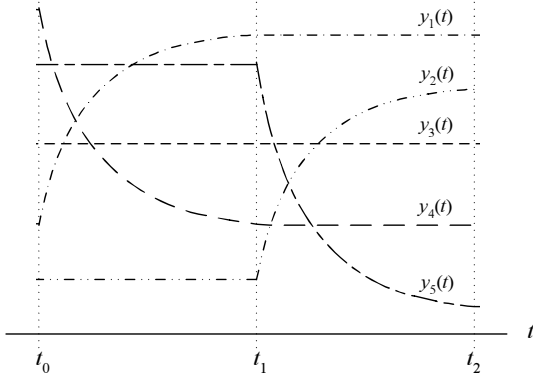


Fig.2. Active and latent components

2 Multi-Rate Runge-Kutta Methods

2.1 Definition

Let us consider two Runge-Kutta (RK) methods [5], [8], that can but do not have to be the same, expressed by their Butcher tableaus (b, A, c) and $(\bar{b}, \bar{A}, \bar{c})$, for integrating y_A and y_L , respectively. The resulting multi-rate Runge-Kutta (MRK) method for the numerical solution of (2) is defined as follows [6]:

- the active components y_A are given by

$$y_A(t_0 + (\lambda + 1)h) \approx y_{A,\lambda+1} = y_{A,\lambda} + h \sum_{i=1}^s b_i k_{A,i}^\lambda, \\ \lambda = 0, 1, \dots, m-1,$$

$$k_{A,i}^\lambda = f_A \left(y_{A,\lambda} + h \sum_{j=1}^s a_{ij} k_{A,j}^\lambda, \tilde{Y}_{L,i}^\lambda \right), \quad i = 1, 2, \dots, s,$$

where $\tilde{Y}_{L,i}^\lambda \approx y_L(t_0 + (\lambda + c_i)h)$ and $c_i = \sum_{j=1}^s a_{ij}$,

- the latent components y_L are given by

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

$$k_{L,i} = f_L \left(\tilde{Y}_{A,i}, y_{L,0} + H \sum_{j=1}^{\bar{s}} \bar{a}_{ij} k_{L,j} \right), \quad i = 1, 2, \dots, \bar{s},$$

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

As we can see the coupling between active and latent subsystems is performed by the intermediate stage values $\tilde{Y}_{A,i}$ and $\tilde{Y}_{L,i}$. There are several strategies for computing this values, like for example the ones suggested by Günther and Rentrop in [3] and [4], which are based in interpolation and/or extrapolation techniques. However, the algorithms studied and tested in this paper are the ones more recently proposed by Kværnø and Rentrop: MRKI and MRKII.

2.2 MRKI and MRKII Algorithms

In the MRKI algorithm the intermediate stage values $\tilde{Y}_{A,i}$ and $\tilde{Y}_{L,i}$ are obtained by

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

and

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

We must note that the $\tilde{Y}_{A,i}$ values are here somehow obtained by extrapolation from the first microstep.

MRKII is a more robust algorithm, where the $\tilde{Y}_{L,i}$ values are given just as in MRKI, but where the $\tilde{Y}_{A,i}$ values are obtained as follows:

$$v_0 = y_{A,0}, \quad k_{L,1} = f_L(y_{A,0}, y_{L,0})$$

for $i = 1, 2, \dots, \bar{s}-1$

for $\lambda = \bar{c}_i m, \dots, \bar{c}_{i+1} m - 1$

$$l_{\lambda+1} = f_A \left(v_\lambda, y_{L,0} + h \sum_{j=1}^i \xi_{ij}(\lambda) k_{L,j} \right)$$

$$v_{\lambda+1} = v_\lambda + h \cdot l_{\lambda+1}$$

end

$$\tilde{Y}_{A,i+1} = y_{A,0} + H \sum_{\lambda=1}^{\bar{c}_{i+1} m} \bar{\gamma}_{i+1,\lambda} l_\lambda$$

$$k_{L,i+1} = f_L \left(\tilde{Y}_{A,i+1}, y_{L,0} + H \sum_{j=1}^i \bar{a}_{i+1,j} k_{L,j} \right)$$

end

As we can see, in both MRK algorithms the (b, A, c) and $(\bar{b}, \bar{A}, \bar{c})$ classical Runge-Kutta methods are connected by coupling coefficients. In the MRKI we have γ_{ij} , $\bar{\gamma}_{ij}$ and $\eta_j(\lambda)$, while in the MRKII we have γ_{ij} , $\eta_j(\lambda)$, $\bar{\gamma}_{i\lambda}$ and $\xi_{ij}(\lambda)$. Obviously all these coefficients must be carefully chosen, in way to guarantee a certain order, or to retain the order of the (b, A, c) and $(\bar{b}, \bar{A}, \bar{c})$, if they are of the same order. There are several ways to

derive order conditions for MRK methods, but the most appropriate seems to be rewriting them as partitioned Runge-Kutta (PRK) methods. By applying the already established order theory for PRK [5], order conditions for MRKI and MRKII may be found. All details of this order study can be seen in [7] or [9].

2.3 The Bogacki-Shampine RK Method

In both MRKI and MRKII algorithms described above we have considered for (b, A, c) and $(\bar{b}, \bar{A}, \bar{c})$ the same method: the Bogacki-Shampine embedded Runge-Kutta method [1]. Its Butcher tableau is

0	0			
1/2	1/2	0		
3/4	0	3/4	0	
1	2/9	3/9	4/9	0
b_i	2/9	3/9	4/9	
\hat{b}_i	7/24	6/24	8/24	3/24

Table 1. Bogacki-Shampine (2)3

and it's not difficult to verify that with b_i this explicit method is of order three, while with \hat{b}_i it is only of order two, in spite of having four stages (one extra stage is added for error estimation).

In this case one possible set of coupling coefficients that satisfies the order conditions up to three in MRKI and MRKII is presented in Table 2.

$\gamma_{21} = \frac{1}{2}$	$\gamma_{31} = \frac{3}{4} \left(1 - \frac{1}{m}\right)$	$\gamma_{32} = \frac{3}{4m}$
$\eta_1 = \left(-\frac{1}{m} + \frac{3}{2} - \frac{m}{4}\right) \lambda - \frac{1}{2m} \lambda^2$		
$\eta_2 = \left(\frac{1}{m} - \frac{3}{2} + \frac{3m}{4}\right) \lambda - \frac{1}{2m} \lambda^2$		
$\eta_3 = \left(1 - \frac{m}{2}\right) \lambda + \frac{1}{m} \lambda^2$		
$\bar{\gamma}_{21} = \frac{1}{2}$	$\bar{\gamma}_{31} = \frac{3(1-m)}{4}$	$\bar{\gamma}_{32} = \frac{3m}{4}$
$\bar{\gamma}_{3,m/2} = -\frac{5}{4m}$	$\bar{\gamma}_{3,3m/4} = \frac{13}{4m}$	$\bar{\gamma}_{i,\lambda} = \frac{1}{m}$
$\xi_{ij}(\lambda) = \frac{\lambda}{i}$		

Table 2. Coupling coefficients

We must observe that this choice of coupling coefficients is not unique. Other choices are possible. It is so because the number of coefficients is larger

than the number of order conditions (omitted here for brevity).

3 Stability

Numerical stability properties of various multi-rate schemes have been discussed by several authors. Unfortunately, most of these discussions (including the one presented in [6]) are not very detailed, nor very conclusive, and until now a concise theory is missing.

Beyond the study of the influence of the stiffness of the problem or the coupling between the active and latent subsystems, the main goal of this section is to compare the stability properties of MRKI with MRKII.

3.1 Multi-Rate Test Problem

The absolute stability properties of an integration method are usually studied by applying the method to the scalar test equation $y' = \alpha y$, with $\alpha \in \mathbb{C}^-$. Doing so for a standard Runge-Kutta method, the solution after one step h is given by

$$y_1 = R(h\alpha)y_0,$$

where $R(h\alpha)$ is the stability function of the method. The method is stable if and only if $|R(h\alpha)| < 1$ [8]. However, multi-rate schemes require at least two components, so the linear differential system

$$\begin{bmatrix} \dot{y}_A \\ \dot{y}_L \end{bmatrix} = \mathcal{A} \begin{bmatrix} y_A \\ y_L \end{bmatrix}, \quad \mathcal{A} = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \in \mathbb{R}^{2 \times 2}, \quad (3)$$

might be an appropriate test problem. If the assumptions

$$\alpha_{11}, \alpha_{22} < 0, \quad \gamma = \frac{\alpha_{12}\alpha_{21}}{\alpha_{11}\alpha_{22}} < 1,$$

are satisfied, then no extra conditions are required to ensure that both eigenvalues of \mathcal{A} have negative real parts. The parameter γ can be seen as a measure for the coupling between the equations and we also define $\kappa = \alpha_{11} / \alpha_{22}$ as a measure for the stiffness of the system.

The numerical solution of (3) performed by the algorithms MRKI or MRKII after one compound step (1 macrostep H for y_L and m microsteps h for y_A) can be expressed by

$$\begin{bmatrix} y_{A,m} \\ y_{L,1} \end{bmatrix} = \mathcal{K} \begin{bmatrix} y_{A,0} \\ y_{L,0} \end{bmatrix},$$

where \mathcal{K} is a matrix that depends on H, h, \mathcal{A} and the version of the MRK algorithm. It doesn't depend on the initial condition and for fixed step lengths H

and h it remains constant throughout the integration process. Thus, for example, from $\mathcal{K} y_0 = y_1$

$$\begin{cases} \mathcal{K}_{11}y_{A,0} + \mathcal{K}_{12}y_{L,0} = y_{A,m} \\ \mathcal{K}_{21}y_{A,0} + \mathcal{K}_{22}y_{L,0} = y_{L,1} \end{cases}$$

and $\mathcal{K} y_1 = y_2$

$$\begin{cases} \mathcal{K}_{11}y_{A,m} + \mathcal{K}_{12}y_{L,1} = y_{A,2m} \\ \mathcal{K}_{21}y_{A,m} + \mathcal{K}_{22}y_{L,1} = y_{L,2} \end{cases}$$

we have

$$\begin{cases} \mathcal{K}_{11}y_{A,0} + \mathcal{K}_{12}y_{L,0} + 0 + 0 = y_{A,m} \\ 0 + 0 + \mathcal{K}_{21}y_{A,0} + \mathcal{K}_{22}y_{L,0} = y_{L,1} \\ \mathcal{K}_{11}y_{A,m} + \mathcal{K}_{12}y_{L,1} + 0 + 0 = y_{A,2m} \\ 0 + 0 + \mathcal{K}_{21}y_{A,m} + \mathcal{K}_{22}y_{L,1} = y_{L,2} \end{cases},$$

that is to say,

$$\underbrace{\begin{bmatrix} y_{A,0} & y_{L,0} & 0 & 0 \\ 0 & 0 & y_{A,0} & y_{L,0} \\ y_{A,m} & y_{L,1} & 0 & 0 \\ 0 & 0 & y_{A,m} & y_{L,1} \end{bmatrix}}_{\mathcal{M}} \begin{bmatrix} \mathcal{K}_{11} \\ \mathcal{K}_{12} \\ \mathcal{K}_{21} \\ \mathcal{K}_{22} \end{bmatrix} = \begin{bmatrix} y_{A,m} \\ y_{L,1} \\ y_{A,2m} \\ y_{L,2} \end{bmatrix}.$$

Now, if $\det(\mathcal{M}) \neq 0$ it is possible to find the entries \mathcal{K}_{11} , \mathcal{K}_{12} , \mathcal{K}_{21} and \mathcal{K}_{22} . The method is stable if and only if the spectral radius $\rho(\mathcal{K})$ of the matrix \mathcal{K} satisfies $\rho(\mathcal{K}) < 1$.

Step sizes h and H are chosen to ensure stability for the uncoupled system ($\alpha_{12} = \alpha_{21} = 0$), i.e., to ensure that the stability functions $R_A(h\alpha_{11})$ and $R_L(H\alpha_{22})$ satisfy the conditions $|R_A(h\alpha_{11})| < 1$ and $|R_L(H\alpha_{22})| < 1$. In the case of the Bogacki-Shampine method that means [8] $-2.54 < h\alpha_{11} < 0$ and $-2.54 < H\alpha_{22} < 0$, that is to say, $H < 2.54$ and $m \geq \kappa$ if, with no loss of generality, we make $\alpha_{22} = -1$. The question is to know how the coupling γ between the two systems affects the stability of the MRK and in what way it depends on κ , H and m .

3.2 Experimental Results

Experimental results obtained computationally in *MATLAB*[®] are shown in Fig.3, where we have plots of some stability regions for $\kappa = 1$, $\kappa = 10$ and $\kappa = 30$. The methods are stable below the boundaries and unstable above and as it can be seen these stability regions become smaller with increasing κ , $|\gamma|$ and m . We have also tested other values of κ larger than 30 and we obtained similar results with smaller regions (we just omitted them here for brevity). From the above we conclude that increasing the stiffness of the system, the coupling between the parts, or the number of microsteps, we force the methods to use a smaller macrostep H . We can also

see in Fig.3 that MRKII is more robust than MRKI, since it has larger stability regions. According to the performance of both algorithms, this result is more coherent than the one presented by Kværnø in [6].

4 Technical Details

Before presenting our numerical results for the simulation of an electronic test example with MRKI and MRKII, we will first introduce some technical details used for the code implementation of these two algorithms: stiffness detection, step size control and partitioning strategy.

4.1 Stiffness Detection

The Bogacki-Shampine embedded Runge-Kutta method used in both MRK algorithms is an explicit method. As we know, explicit codes applied to stiff problems are not very efficient. So, in order to avoid that MRKI and MRKII waste too much effort when encountering stiffness, it is important that the code be equipped with an automatic detecting stiffness structure. Our strategy is presented in the following.

The local error of the Bogacki-Shampine method can be estimated by

$$err = h \sum_{i=1}^4 (b_i - \hat{b}_i) k_i = h \sum_{i=1}^4 d_i k_i. \quad (4)$$

On the other hand, in a linear problem the numerical solution after one step is given by

$$y_1 = R(z) y_0,$$

where $R(z)$ is the stability function of the method (b, A, c) . In the same way we can say that

$$\hat{y}_1 = \hat{R}(z) y_0,$$

with $\hat{R}(z)$ the stability function of the method (\hat{b}, A, c) . So, the error estimate can be given by

$$err = y_1 - \hat{y}_1 = [R(z) - \hat{R}(z)] \cdot y_0,$$

that is to say,

$$err = E(z) \cdot y_0,$$

with

$$E(z) = R(z) - \hat{R}(z).$$

Let us consider now a new embedded Runge-Kutta method, replacing \hat{b} by \tilde{b} in the Bogacki-Shampine scheme. In a similar way, we have

$$\widetilde{err} = y_1 - \tilde{y}_1 = [R(z) - \tilde{R}(z)] \cdot y_0,$$

i.e., a new error estimate defined by

$$\widetilde{err} = \tilde{E}(z) \cdot y_0, \quad (5)$$

with

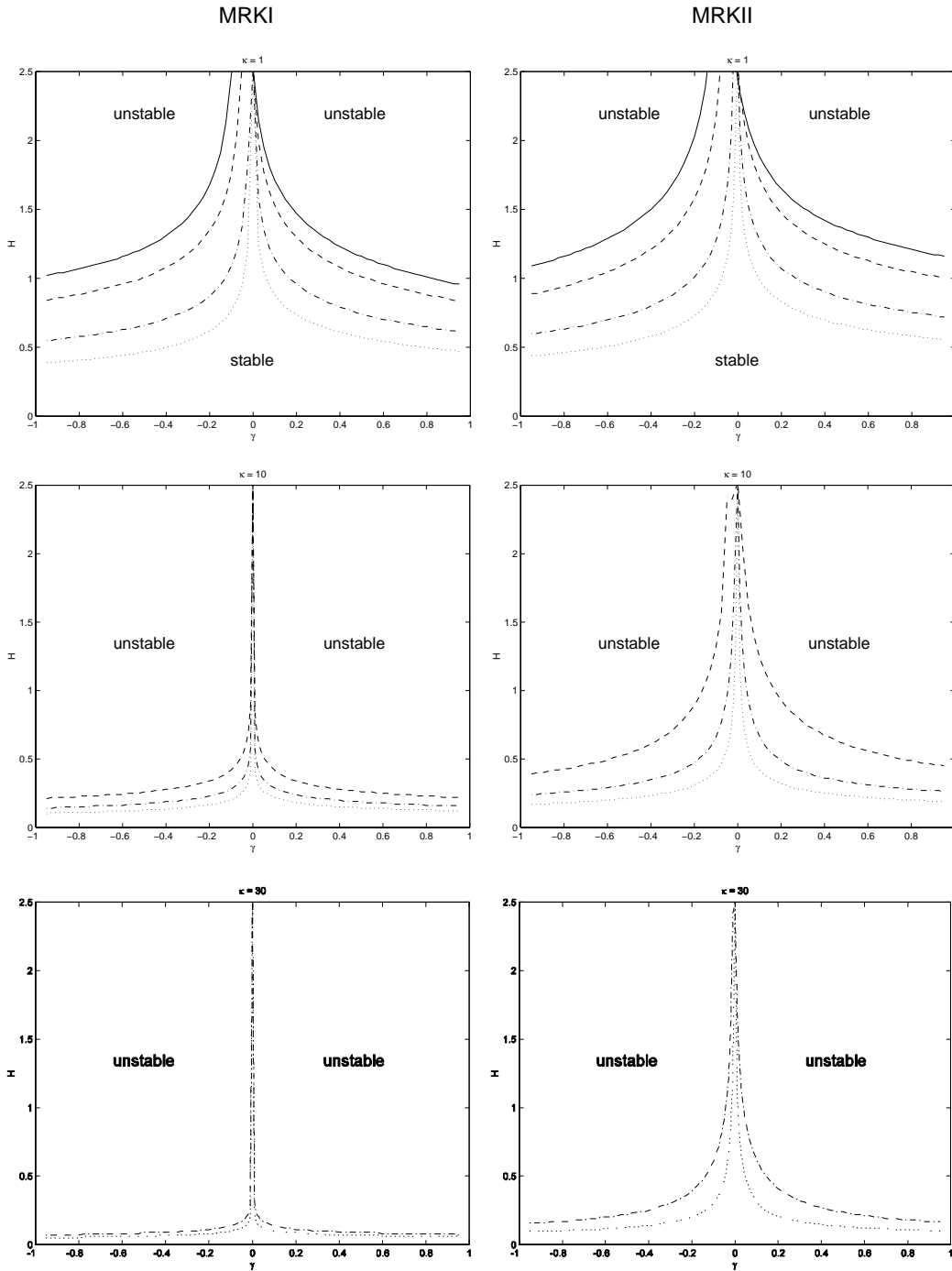


Fig.3. Stability regions: — $m=8$, --- $m=12$, ··· $m=36$, ···· $m=100$

$$\tilde{E}(z) = R(z) - \tilde{R}(z).$$

The key idea is to choose the vector \tilde{b} so that the method (\tilde{b}, A, c) is of order one, meaning that $\tilde{err} = O(h^2)$, when $h \rightarrow 0$,

and also that the condition

$$|\tilde{E}(z)| \leq \theta |E(z)|, \quad \theta < 1$$

is satisfied in the boundary of the stability region \mathfrak{R} . If it is so, then we will have $|\tilde{err}| \gg |err|$ for small

step sizes h , i.e., step sizes which are determined by accuracy requirements (when the problem is not stiff) and $|\tilde{err}| < |err|$ when we are working near of the boundary of the stability region (the problem is possible stiff). A computer search gave us the following values for \tilde{b} :

$$\tilde{b}^T = [0.2817 \quad 0.2800 \quad 0.4353 \quad 0.0030].$$

With these values $\tilde{E}(z)/E(z) \approx 0.3$ in almost the whole boundary of \mathfrak{R} , as we can see in Fig.4.

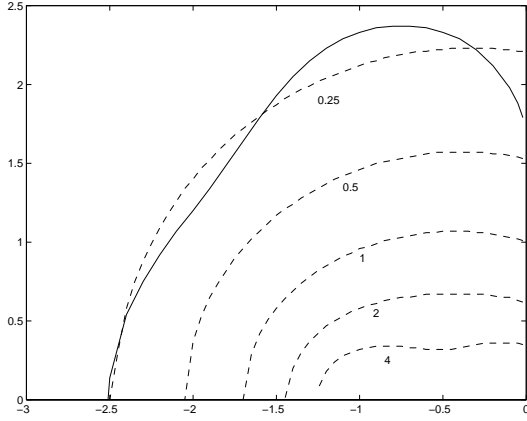


Fig.4. Stability region \mathfrak{R} and contour lines $\tilde{E}(z)/E(z) = 0.25, 0.5, 1, 2$ and 4

4.2 Step Size Control

We want to have a code which automatically adjusts its step sizes in order to achieve a prescribed tolerance for the error. According to (4) the error estimates for the active and latent components in MRKI and MRKII algorithms are given by

$$err_{A,\lambda+1} = h \sum_{i=1}^4 d_i k_{A,i}^\lambda, \quad err_{L,1} = H \sum_{i=1}^4 d_i k_{L,i}, \quad (6)$$

where

$$k_{A,4}^\lambda = f_A \left(y_{A,\lambda+1}, y_{L,0} + h \sum_{j=1}^3 (\eta_j (\lambda + 1)) k_{L,j} \right)$$

and

$$k_{L,4} = f_L (y_{A,m}, y_{L,1}).$$

Selection of new step lengths h and H is performed at the end of each macrostep, and the whole strategy for automatic step size control throughout the numerical integration process goes as follows: based on the error estimates (6) a new step length h_{prop} is proposed for each component, defined by

$$h_{prop} = 0.8 \cdot \left(\frac{Tol}{|err|} \right)^{\frac{1}{p+1}} \cdot h_{old},$$

where Tol is the desired error tolerance, p is the order of the method and $h_{old} = h$ (active components) or $h_{old} = H$ (latent components). The value 0.8 is just a safety factor so that the error will be acceptable the next time with high probability. Additionally, alternative error estimates (5) are also computed for each component and if $|err| < |err|$ the component is marked stiff. A new macrostep length H_{new} is then obtained by

$$H_{new} = \min \left(\min \left(\underbrace{h_{prop}}_{stiff} \right), 0.5 \cdot \max \left(\underbrace{h_{prop}}_{all} \right) \right),$$

but not allowed to increase too fast nor to decrease too fast,

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

to prevent a step size control with high zigzag. A new microstep length h_{new} is initially given by

$$h_{new} = \min \left(\underbrace{h_{prop}}_{all} \right),$$

and then the value of

$$m = H_{new} / h_{new}$$

must be rounded to the nearest integer towards plus infinity (in MRKI), or to the nearest multiple of four towards plus infinity (in MRKII). The new microstep length h_{new} is finally found by

$$h_{new} = H_{new} / m.$$

4.3 Partitioning Strategy

The partitioning of the differential system into the active and latent subsystems is done according to the following: each component that satisfies the condition $h_{prop} \geq H_{new}$ will be treated as latent on the next iteration; the remaining ones will be treated as actives.

5 Sample Application

5.1 Electronic Pulse Generator

In order to test the performance and the efficiency of our multi-rate algorithms an electronic pulse generator with MOSFETs was simulated with MRKI and MRKII. The schematic of this circuit is shown in Fig.5.

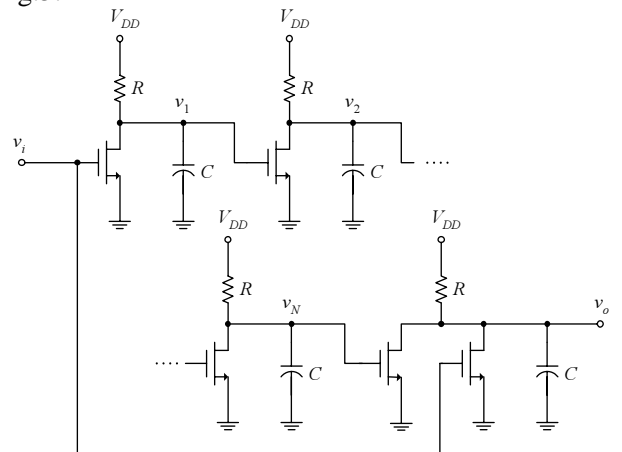


Fig.5. Pulse generator

This kind of circuit is commonly found in digital systems and it generates a positive pulse on its output when detects a transition from state 1 (high) to state 0

(low) on its input. It is constituted by an odd number N of logical inverters connected in chain, followed by a NOR gate. The output of each inverter is the input of the next inverter and the charging and discharging of the capacitors C produces delays in the run time of the signal. Thus, v_N is the logical negation of a phase shifted version of the input v_i , and the output v_o is the logical NOR between v_i and v_N . For a better understanding let us look at the waveform chart illustrated in Fig.6. The input v_i is a Heaviside (step) function where the transition from state 1 to 0 occurs at some time t . v_N is also a step function, but due to the signal propagation delay the transition from one state to another occurs at some time $t' > t$. The result is a positive output pulse v_o , in the interval $[t, t']$.

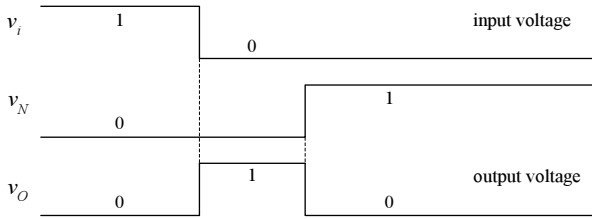


Fig.6. Pulse generator waveform chart

5.2 Mathematical Model

Electronic circuit elements are characterized by equations which relate the voltage v across them to the current i through them (terminal voltage and current). In this case we have resistors R , whose terminal current and voltage are proportional

$$i = \frac{v}{R} \text{ (Ohm's law),}$$

capacitors C , which are characterized by

$$i = C \frac{dv}{dt}$$

and MOSFETs (metal oxide semiconductor field effect transistors). For these we chose a simple non dynamic model, a voltage controlled current source. Its characteristics are modeled in the following way:

$$i_{GS} = i_{GD} = 0, \text{ G - gate, D - drain, S - source,}$$

$$i_{DS} = K \cdot g_{NL}(v_G, v_D, v_S),$$

where g_{NL} is a non linear function defined by

$$g_{NL}(v_G, v_D, v_S) = \left(\max(v_G - v_S - V_T, 0) \right)^2 - \left(\max(v_G - v_D - V_T, 0) \right)^2$$

with $K = 2 \cdot 10^{-4} \text{ AV}^{-2}$ and $V_T = 1 \text{ V}$.

The mathematical model of an electronic circuit (a network made up of interconnections of many circuit elements and generators) is based on Kirchhoff's

current and voltage laws. Kirchhoff's current law tells us that the net current into any node is zero. Kirchhoff's voltage law says that the net sum of the voltage drops around any closed loop is zero.

Let us begin by applying the current law to node v_1 (output of the first inverter) of the pulse generator. We will have

$$i_R = i_C + i_{DS}$$

or, equivalently,

$$\frac{V_{DD} - v_1}{R} = C \frac{dv_1}{dt} + i_{DS}.$$

Now, once $v_G = v_i$, $v_D = v_1$ and $v_S = 0$, we obtain

$$\frac{V_{DD} - v_1}{R} = C \frac{dv_1}{dt} + K \cdot g_{NL}(v_i, v_1, 0)$$

that can be rewritten as

$$\frac{dv_1}{dt} = 10^{-9} \left[\frac{V_{DD} - v_1}{RC} - \frac{K}{C} \cdot g_{NL}(v_i, v_1, 0) \right]. \quad (6)$$

The reason why the scale factor 10^{-9} was intentionally included is because the run time of a digital signal is usually measured in nanoseconds. Similarly, for the next $N-1$ nodes v_k we can write

$$\frac{dv_k}{dt} = 10^{-9} \left[\frac{V_{DD} - v_k}{RC} - \frac{K}{C} \cdot g_{NL}(v_{k-1}, v_k, 0) \right], \quad (7)$$

$$k = 2, \dots, N.$$

Finally, applying the Kirchhoff's current law to node v_o (output of the NOR gate) we have

$$\frac{dv_o}{dt} = 10^{-9} \left[\frac{V_{DD} - v_o}{RC} - \frac{K}{C} \cdot g_{NL}(v_N, v_o, 0) - \frac{K}{C} \cdot g_{NL}(v_i, v_o, 0) \right]. \quad (8)$$

Equations (6), (7) and (8) make up the differential system that models our pulse generator.

5.3 Numerical Simulation Results

The circuit with $N = 51$ inverters was simulated in *MATLAB*[®] from $t = 0$ to $t = 40 \text{ ns}$, for an input v_i transition at $t = 1 \text{ ns}$. The numerical solution v_o is shown in Fig.7 and the overall results of this simulation are presented in Table 3. In all our tests we have considered $V_{DD} = 5 \text{ V}$, $R = 4.7 \text{ k}\Omega$ and $C = 0.2 \text{ pF}$.

	time ¹ (sec)	number of steps			error in v_o	
		rejected	macro	micro	$\ \cdot \ _{\infty}$	$\ \cdot \ _{L^2}$
MRKI	1.37	58	142	406	0.0048	0.0038
MRKII	2.15	60	135	616	0.0034	0.0029

Table 3. Numerical simulation results

¹ Computation time (AMD Athlon 1.8 GHz, 256MB RAM).

In order to test the accuracy of the MRK methods a reference solution was achieved by numerically solving the ODE system (6)-(8) via classical time-step integration, using a classical Runge-Kutta method of higher order, with an extremely small time step.

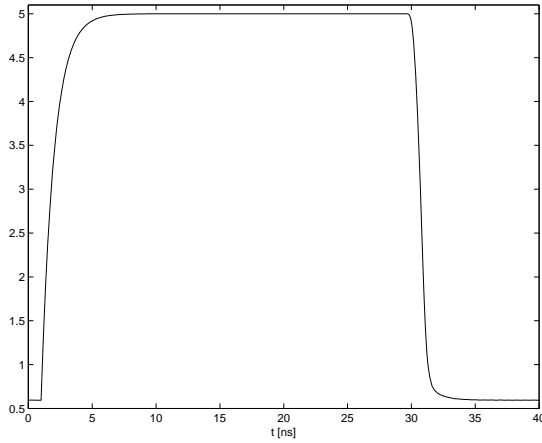


Fig.7. Numerical solution v_O

6 Conclusions

After several simulation tests with different quantities of logical inverters and different integration intervals, we can say that in general both MRK methods show a good performance when solving our sample circuit. However, the MRKII algorithm leads to an increase of the computational work, once the total time for obtaining the numerical solution was in all cases bigger than in MRKI. In Section 3 we saw that the MRKII was a more stable method, nevertheless this stability gain implies a consequent loss of computational speed. So, due to our opinion the MRKII algorithm must be chosen over MRKI only in stiff problems or when the coupling from the active to the latent part is strong.

References:

- [1] P. Bogacki and L. Shampine, A (2)3 pair of Runge-Kutta formulas, *Applied Math. Letters*, No.2, 1989, pp. 1-9.
- [2] M. Günther, A. Kværnø and P. Rentrop, Multirate partitioned Runge-Kutta methods, *BIT*, Vol.3, No.41, 2001, pp. 504-514.
- [3] M. Günther and P. Rentrop, Multirate ROW methods and latency of electric circuits, *Applied Numerical Mathematics*, No.13, 1993, pp. 83-102.
- [4] M. Günther and P. Rentrop, Partitioning and multirate strategies in latent electric circuits, *International Series of Numerical Mathematics*, No.117, 1994, pp. 33-60.

- [5] E. Hairer, S. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer-Verlag, Berlin, 1987.
- [6] A. Kværnø, Stability of multirate Runge-Kutta schemes, *International Journal of Differential Equations and Applications*, No.1A, 2000, pp. 97-105.
- [7] A. Kværnø and P. Rentrop, Low order multirate Runge-Kutta methods in electric circuit simulation, *IWRMM Universität Karlsruhe*, Preprint No.99/1, 1999.
- [8] J. Lambert, *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*, John Wiley & Sons, West Sussex, 1991.
- [9] J. Oliveira, *Métodos Multi-Ritmo na Análise e Simulação de Circuitos Eletrônicos não Lineares*, Master Thesis, Department of Mathematics, University of Coimbra, 2004.