Reduced Order Multirate Simulation of Circuits

M.W.F.M. Bannenberg[1,2], A. Ciccazzo[2] M. Günther[1].

Abstract

This document presents a benchmark case for the application of reduced order multirate schemes in a MATLAB environment. Starting with the problem formulation and then a synopsis of the mathematical techniques applied in the benchmark a test case is discussed. Then the numerical approach to the simulation implementation is outlined, and finally a numerical experiment is shown verifying the validity of this benchmark case.

1 Introduction

In integrated circuit design, there are a significant number of design possibilities under which the internal components need to be guaranteed to work. This leads to a whole range of explorations to ensure sound functionality of the design. These explorations are performed by numerical simulations of the circuits mathematical model. Due to the ever increasing number of components, and thus the degrees of freedom in the model, the required simulation times may become prohibitively large.

Besides the sheer number of components inside the integrated circuit, a large contribution to the complexity of the mathematical model originates form the method of deriving these equations. As these models grow, generating a state-space model with a minimal set of unknowns cannot be generated in an automatic way. Therefore, the mathematical models have to be derived through use of algorithmic analysis. This automation comes at a cost. The resulting system of differential-algebraic equations (DAE) is numerically harder to solve, and may contain redundant network variables.

To decrease these ever increasing simulation costs a multitude of different approaches have been proposed in the past decades. For instance, the redundancy originating from the network analysis can be exploited by using model order reduction. This technique aims to solve a model of reduced size that still approximates the solution of the original model. Furthermore, the large original system can be partitioned into subsystems which each have their own characteristic rate of evolution through time. This property is capitalised upon by using multirate (MR) time integration. This paper is specifically aimed at the combination of the two previously mentioned techniques, and provides an overview from definition to implementation.

[1]IMACM, Chair of Applied Mathematics and Numerical Analysis (AMNA), Bergische Universität Wuppertal, Gaußstraße 20, 42119 Wuppertal, Germany
e-mail:bannenberg@uni-wuppertal.de

[2]STMicroelectronics, Str. Primsole 50, Catania, Italy.

2 Problem Formulation

Consider the following coupled system of two semi explicit DAE systems, where the subscripts $\{F, S\}$ indicate a fast or slow time-scale, respectively, and independent transient sources have been omitted for notational convenience:

$$\frac{d}{dt} y_{\mathrm{F}} = f_{\mathrm{F}}(t, y_{\mathrm{F}}, z_{\mathrm{F}}, y_{\mathrm{S}}, z_{\mathrm{S}}), \quad y_{\mathrm{F}}(t_0) = y_{F_0}, \tag{1a}$$

$$0 = g_{\mathrm{F}}(t, y_{\mathrm{F}}, z_{\mathrm{F}}, y_{\mathrm{S}}, z_{\mathrm{S}}), \quad z_{\mathrm{F}}(t_0) = z_{F_0}, \tag{1b}$$

$$\frac{d}{dt} y_{\mathrm{S}} = f_{\mathrm{S}}(t, y_{\mathrm{F}}, z_{\mathrm{F}}, y_{\mathrm{S}}, z_{\mathrm{S}}), \quad y_{\mathrm{S}}(t_0) = y_{S_0}, \tag{1c}$$

$$0 = g_{\mathrm{S}}(t, y_{\mathrm{F}}, z_{\mathrm{F}}, y_{\mathrm{S}}, z_{\mathrm{S}}), \quad z_{\mathrm{S}}(t_0) = z_{S_0}, \tag{1d}$$

with the functions $f_A : \mathbb{R} \times \mathbb{R}^a \times \mathbb{R}^b \times \mathbb{R}^c \times \mathbb{R}^d \to \mathbb{R}^a$, with $A \in \{\mathrm{F}, \mathrm{S}\}$, where $\{a, b, c, d\} \in \mathbb{N}$ are the respective dimensions, and equivalent definitions for g_A. Consistent initial conditions are assumed, which means that Equations (1b) and (1d) are satisfied at initial time t_0. The quantities $y_{\{F,S\}} : I \to \mathbb{R}^{\{a,b\}}$ and $z_{\{F,S\}} : I \to \mathbb{R}^{\{c,d\}}$ denote the differential and algebraic variables defined on the time interval $[t_0, t_1] = I$. Both subsystems and the joint system are guaranteed to be index-1 by the assumption that the Jacobians

$$\frac{\partial g_{\mathrm{F}}}{\partial z_{\mathrm{F}}}, \frac{\partial g_{\mathrm{S}}}{\partial z_{\mathrm{S}}} \text{ and } \begin{pmatrix} \frac{\partial g_{\mathrm{F}}}{\partial z_{\mathrm{F}}} & \frac{\partial g_{\mathrm{F}}}{\partial z_{\mathrm{S}}} \\ \frac{\partial g_{\mathrm{S}}}{\partial z_{\mathrm{F}}} & \frac{\partial g_{\mathrm{S}}}{\partial z_{\mathrm{S}}} \end{pmatrix} \text{ are invertible} \tag{2}$$

in the neighbourhood of the solution of the system. From this assumption the algebraic variables $z_{\{F,S\}}$ can be solved locally by using the implicit function theorem

$$z_F = G_{t,F}(y_{\mathrm{F}}, z_S, y_{\mathrm{S}}), \tag{3a}$$

$$z_S = G_{t,S}(y_{\mathrm{F}}, z_F, y_{\mathrm{S}}), \tag{3b}$$

where the second z subscript is the opposite of the first z subscript. The partition of the system into subsystems can originate from different physical systems, such as temperature diffusion and electric currents. However, differences in time scale can also be identified by different orders of time derivatives. Here the partition is considered to be fixed during the time integration.

3 The Reduced Order Multirate Method

In this section a brief overview of the mathematical aspects of the reduced order multirate method is presented. For a more in depth description see [2, 1, 3].

3.1 Maximum Entropy Snapshot Sampling

Let m and n be positive integers and $m \gg n > 1$. Define a finite sequence $X = (x_1, x_2, ..., x_n)$ of numerically obtained states $x_j \in \mathbb{R}^m$ at time instances

$t_j \in \mathbb{R}$, with $j \in \{1, 2, ..., n\}$, of a dynamical system governed by either ODEs or DAEs. Provided a probability distribution p of the states of the system, the second-order Rényi entropy of the sample X is

$$H_p^{(2)}(X) = -\log \sum_{j=1}^{n} p(x_j)^2 = -\log \mathbb{E}(p(X)), \tag{4}$$

with $\mathbb{E}(p(X))$ the expected value of the probability distribution p with respect to p itself. When n is large enough, according to the law of large numbers, the average of $p_1, p_2, ..., p_n$ almost surely converges to their expected value,

$$\frac{1}{n} \sum_{j=1}^{n} p(x_j) \to \mathbb{E}(p(X)) \quad \text{as } n \to \infty, \tag{5}$$

thus each $p(x_j)$ can be approximated by the sample's average sojourn time or relative frequency of occurrence. To obtain this frequency of occurrence, consider a norm $\|\cdot\|$ on $\mathbb{R}^m$. Then the notion of occurrence can be translated into a proximity condition. In particular, for each $x_j \in \mathbb{R}^m$ define the open ball that is centred at x_j and whose radius is $\epsilon > 0$,

$$B_\epsilon(x) = \{y \in \mathbb{R}^m \mid \|x - y\| < \epsilon\}, \tag{6}$$

and introduce the characteristic function with values

$$\chi_i(x) = \begin{cases} 1, & \text{if } x \in B_\epsilon(x_i), \\ 0, & \text{if } x \notin B_\epsilon(x_i). \end{cases} \tag{7}$$

Under the aforementioned considerations, the entropy of X can be estimated by

$$\hat{H}_p^{(2)}(X) = -\log \left(\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \chi_i(x_j) \right). \tag{8}$$

Provided that the limit of the evolution of $\hat{H}_p^{(2)}$ exists, for n large enough, and measures the sensitivity of the evolution of the system itself [4, §6.6], a reduced sequence $X_\mathrm{r} = (\bar{x}_{j_1}, \bar{x}_{j_2}, \ldots, \bar{x}_{j_r})$, with $r \leq n$, is sampled from X, by requiring that the entropy of X_r is a strictly increasing function of the index $k \in \{1, 2, \ldots, r\}$ [6]. The state vector $\bar{x}_{j_k}$ added to sampled snapshot space is the average value of all states in the selected ϵ-ball. A reduced basis is then generated from X_r with any orthonormalization process.

The Estimation of ϵ: The open ball parameter ϵ, which is directly responsible for the degree of reduction within the MESS framework, can be chosen arbitrarily, much like the number of selected basis vectors provided by a POD approach. For a ballpark estimate of this parameter the following optimisation approach is provided. The quantity within the logarithm in the entropy estimate (8) is often referred to as the sample's correlation sum and can be written as

$$C_\epsilon = \frac{1}{n^2} \|R_\epsilon\|_\mathrm{F}^2, \tag{9}$$

with $R_\epsilon \in \{0,1\}^{n\times n}$ being the recurrence matrix whose entries are unity, when $\|x_i - x_j\| < \epsilon$, and $\|\cdot\|_{\mathrm{F}}^2$ being the Frobenius norm. In terms of probability theory, C_ϵ is a cumulative distribution function of ϵ, and hence, its derivative $\mathrm{d}C_\epsilon/\mathrm{d}\epsilon$ is the associated probability density function of ϵ. A commonly justified hypothesis is that the correlation sum scales as ϵ^D [7, Chapter 1], with $D \geq 0$ being the so-called correlation dimension of the manifold that is formed in $\mathbb{R}^m$ by the terms of X. Under this power law assumption, the maximum likelihood estimate [8, Chapter 8] of the correlation dimension is estimated as follows. We find a sample $\{\epsilon_i\}$, with $\epsilon_i \in [0,1]$ for all $i \in \{1,2,\ldots,q\}$, of a random variable E that is sampled according to C_ϵ. Then, the probability of finding a sample in $(\epsilon_i, \epsilon_i + \mathrm{d}\epsilon_i)$ in a trial is

$$\prod_{i=1}^{q} D\epsilon^{D-1}\mathrm{d}\epsilon_i. \tag{10}$$

To calculate the ϵ value for which this expression is maximized, we take the logarithm

$$q \cdot \ln D + (D-1)\sum_{i=1}^{q} \ln \epsilon_i, \tag{11}$$

and note that the maximum of this expression is attained when

$$\frac{q}{D} + \sum_{i=1}^{q} \ln \epsilon_i = 0. \tag{12}$$

This results in the most likely value $D_* = -1/\langle \ln E\rangle$. The value for ϵ_* is then estimated by choosing the ϵ from the sample that produces a quotient that is closest to D^*. Thus ϵ can be estimated by

$$\epsilon_* = \operatorname{argmin}(|D_* - \ln C_\epsilon / \ln \epsilon|). \tag{13}$$

3.2 The Gauß-Newton with approximated tensors method

Unfortunately, a direct application of MESS is not feasible in practice, [9, Section 7.6], therefore a simplified Gauß-Newton with Approximated Tensors (GNAT), equipped with a function-sampling-hyper-reduction scheme is used. Firstly, a direct Galerkin projection may yield an unsolvable reduced system for DAEs. Secondly, the computational effort required to solve this reduced system and the full system is about the same in the nonlinear cases. This is due to the fact that the evaluation costs of the reduced system are not reduced at all because the projection basis will be a dense matrix in general.

Considering a general DAE in the form

$$\dot{\phi}(t,u) + \psi(t,u) = 0, \tag{14}$$

where ϕ and ψ are functions of time t and some state vector u. In the discrete case, we assume that the numerical scheme exactly solves the following nonlinear system for each time step t_i,

$$R(u) = 0, \tag{15}$$

where $u \in \mathbb{R}^N$, u^0 the initial condition and the residual $R : \mathbb{R}^N \to \mathbb{R}^N$. Note that for ease of notation, the relevant time subscripts have been omitted, as this equation is solved for each individual time step. For the reduction of the dimension of Equation (15), a projection is used to search the approximated solution in the incremental affine trial subspace $u^0 + \mathcal{V} \subset \mathbb{R}^N$. Thus $\tilde{u}$ is given by

$$\tilde{u} = u^0 + V_u u_r, \tag{16}$$

where $V_u \in \mathbb{R}^{N \times n_u}$ is the n_u-dimensional projection basis for $\mathcal{V}$, and u_r denotes the reduced incremental vector of the state vector. Now deviating from the direct Galerkin projection process, Equation (16) is substituted into Equation (15). This results in an overdetermined system of N equations and n_u unknowns. Because V_u is a matrix with full column rank, it is possible to solve this system by a minimisation in least-squares sense through

$$\min_{\tilde{u} \in u^0 + \mathcal{V}} R(\tilde{u})_2. \tag{17}$$

This nonlinear least-squares problem is solved by the Gauß-Newton method, leading to the iterative process for $k = 1, ..., K$, solving

$$s^k = \operatorname{argmin}_{a \in \mathbb{R}^{n_u}} J^k V_u a + R^k{}_2, \tag{18}$$

and updating the search value w_r^k with

$$w_r^{k+1} = w_r^k + s^k, \tag{19}$$

where K is defined through a convergence criterion, initial guess w_r^0, $R^k \equiv R(u^0 + V_u w_r^k)$ and $J^k \equiv \frac{\partial R}{\partial u}(u^0, V_u u_r^k)$. Here J^k is the full order Jacobian of the residual at each iteration step k. Since the computation of this Jacobian scales with the original full dimension of Equation (15) this is a computational bottleneck. This bottleneck can be circumvented by the application of hyper reduction methods, for which this paper utilises a gappy data reconstruction method.

Gappy Maximum Entropy Snapshot Sampling: The evaluation of the nonlinear function $R(u_0 + V_u w_r^k)$ has a computational complexity that is still dependent on the size of the full system. To reduce the complexity of this evaluation the gappy MESS procedure, based on gappy POD, is applied. Like the gappy POD approach gappy MESS uses a reduced basis to reconstruct gappy data. However, unlike the gappy POD approach the basis used is now not obtained through POD but by MESS. Gappy MESS starts by defining a mask vector n for a solution state u as

$$n_j = 0 \text{ if } u_j \text{ is missing,}$$
$$n_j = 1 \text{ if } u_j \text{ is known,}$$

where j denotes the j-th element of each vector. The mask vector n is applied point-wise to a vector by $(n, u)_j = n_j u_j$. This sets all the unobserved values to

0. Then, the gappy inner product can be defined as $(x, y)_n = ((n, x), (n, y))$, which is the inner product of the each vector masked respectively. The induced norm is then $(x_n)^2 = (x, x)_n$. Considering the reduction basis obtained by MESS $V_{\text{gap}} = \{v^i\}_{i=1}^r$, now we can construct an intermediate "repaired" full size vector $\tilde{g}$ from a reduced vector g with only r elements by

$$\tilde{g} \approx \sum_{i=1}^{r} b_i v^i, \tag{20}$$

where the coefficients b_i need to minimise an error E between the original and repaired vector, which is defined as

$$E = g - \tilde{g}_n^2, \tag{21}$$

using the gappy norm so that only the original existing data elements in g are compared. This minimisation is done by solving the linear system

$$Mb = f, \tag{22}$$

where

$$M_{ij} = (v^i, v^j)_n, \text{ and } f_i = (g, v^i)_n. \tag{23}$$

From this solution $\tilde{g}$ is constructed. Then the complete vector is reconstructed by mapping the reduced vectors elements to their original indices and filling the rest with the reconstructed values.

3.3 The Reduced System

To incorporate the previous two sections into the partitioned DAE system (1a)-(1d), we first rewrite (1c)-(1d) in a more general DAE form, to have the slow subsystem encapsulated into one equation.

$$\frac{d}{dt} y_{\mathrm{F}} = f_{\mathrm{F}}(t, y_{\mathrm{F}}, z_{\mathrm{F}}, u_{\mathrm{S}}), \quad y_{\mathrm{F}}(t_0) = y_{\mathrm{F}_0}, \tag{24}$$

$$0 = g_{\mathrm{F}}(t, y_{\mathrm{F}}, z_{\mathrm{F}}, u_{\mathrm{S}}), \quad z_{\mathrm{F}}(t_0) = z_{\mathrm{F}_0}, \tag{25}$$

$$\frac{d}{dt}\phi(u_{\mathrm{S}}) = F_{\mathrm{S}}(t, y_{\mathrm{F}}, z_{\mathrm{F}}, u_{\mathrm{S}}), \quad u_{\mathrm{S}}(t_0) = (y_{\mathrm{S}_0}, z_{\mathrm{S}_0})^\top, \tag{26}$$

where $F_{\mathrm{S}} : \mathbb{R} \times \mathbb{R}^a \times \mathbb{R}^b \times \mathbb{R}^{m_{\mathrm{S}}} \to \mathbb{R}^{m_{\mathrm{S}}}$ and $u_{\mathrm{S}} = (y_{\mathrm{S}}, z_{\mathrm{S}})^\top$. Into these equations we incorporate the back projected reduced state $\tilde{u}_{\mathrm{S_r}} = u_{\mathrm{S}}^0 + V_u u_{\mathrm{S_r}}$

$$\frac{d}{dt} y_{\mathrm{F_r}} = f_{\mathrm{F}}(t, y_{\mathrm{F_r}}, z_{\mathrm{F_r}}, \tilde{u}_{\mathrm{S_r}}), \tag{27}$$

$$0 = g_{\mathrm{F}}(t, y_{\mathrm{F_r}}, z_{\mathrm{F_r}}, \tilde{u}_{\mathrm{S_r}}), \tag{28}$$

$$\frac{d}{dt}\phi(\tilde{u}_{\mathrm{S_r}}) = F_{\mathrm{S}}(t, y_{\mathrm{F_r}}, z_{\mathrm{F_r}}, \tilde{u}_{\mathrm{S_r}}). \tag{29}$$

and then, with the Gappy MESS complexity reduction incorporated we obtain

$$\frac{d}{dt}y_{\mathrm{F_r}} = f_{\mathrm{F}}(t, y_{\mathrm{F_r}}, z_{\mathrm{F_r}}, \tilde{u}_{\mathrm{S_r}}), \tag{30}$$

$$0 = g_{\mathrm{F}}(t, y_{\mathrm{F_r}}, z_{\mathrm{F_r}}, \tilde{u}_{\mathrm{S_r}}), \tag{31}$$

$$\frac{d}{dt}\phi(\tilde{u}_{\mathrm{S_r}}) = F_{\mathrm{S_r}}(t, y_{\mathrm{S_r}}, z_{\mathrm{F_r}}, \tilde{u}_{\mathrm{S_r}}). \tag{32}$$

Where $F_{\mathrm{S_r}}$ denotes the function F_{S} solved by the reduced least squares approach. Note that the subscript r denotes a reduction, and not the reduction factor.

4 Numerical integration

Since the set of equations used to describe the electrical circuits is constructed according to the topological structure of the network. This often results in a coupled system of implicit differential and nonlinear equations, or more general a system of differential-algebraic equations (DAEs)

$$f(\dot{x}, x, t) = 0 \text{ with } \det \frac{\partial f}{\partial \dot{x}} \equiv 0. \tag{33}$$

This system may represent ill-posed problems and is in general more difficult to solve numerically than the more standard systems of ordinary differential equations (ODEs).

4.1 Backward Differentiation Formula

Starting from the consistent initial values, the time domain is discretised into time points $t_0, t_1, ..., t_N$, and the solution for each of these time points is approximated by an implicit linear numerical integration formula. A direct approach, as proposed in [11], is by applying backward differentiation formula (BDF) method. This multistep method is applied to a DAE system by using the ϵ-embedding method. Consider a semi-explicit system with dynamical variables y and algebraic variables z,

$$\dot{y} = f(y, z), \tag{34a}$$

$$\epsilon\dot{z} = g(y, z). \tag{34b}$$

then the multistep method gives

$$\sum_{i=0}^{k} \alpha_i y_{n+i} = h \sum_{i=0}^{k} \beta_i f(y_{n+i}, z_{n+i}), \tag{35a}$$

$$\epsilon \sum_{i=0}^{k} \alpha_i z_{n+i} = h \sum_{i=0}^{k} \beta_i g(y_{n+i}, z_{n+i}). \tag{35b}$$

$$\tag{35c}$$

Then by putting $\epsilon = 0$ we obtain

$$\sum_{i=0}^{k} \alpha_i y_{n+i} = h \sum_{i=0}^{k} \beta_i f(y_{n+i}, z_{n+i}), \tag{36a}$$

$$0 = \sum_{i=0}^{k} \beta_i g(y_{n+i}, z_{n+i}). \tag{36b}$$

$$\tag{36c}$$

which enables us to apply this method to a semi-explicit differential algebraic system. However, we want to solve system (**??**), which can be an implicit differential algebraic system. Therefore, the multistep system for an implicit DAE system, $M\dot{x} = f(x)$, is given by

$$M \sum_{i=0}^{k} \alpha_i x_{n+i} = h \sum_{i=0}^{k} \beta_i f(x_{n+i}) \tag{37}$$

In general form, applying Equation (37) to an implicit nonlinear system of DAEs at time step t_n yields

$$f(\frac{1}{h} \sum_{i=0}^{k} \frac{\alpha_i}{\beta_i} x_{n-i}, x_n, t_n) = 0. \tag{38}$$

This gives that the numerical solution of the system is thus reduced to the solution of the system of nonlinear Equations 38. This system is solved iteratively for x_n by applying Newton's method.

4.2 Multirate time-integration

The previously seen integration method is considered a singlerate time-integration method, as it integrates each part of the equation with the same step-size. Opposed to this classical approach there are the multirate time-integration methods, which use a different step-size, or even integration method, for parts of the equations with different dynamical behaviour.

To apply multirate time-integration methods to DAEs it is required that all the subsystems in itself are stable, which means that the DAE-index should be less than or equal to that of the full DAE. As we are partitioning network equations of circuit models, it is known that they are composed of subcircuits or natural phenomena in a hierarchical way. If we partition based on this hierarchy, we can check easily if the subsystems fulfil the index requirements and obtain viable partitions.

As there are several different approaches available to implement multirate time-integration methods, we specify in this subsection which approach is used. As has been shown in [1] and [2] a feasible method is given by the *Coupled-Slowest-First* integration approach coupled with the implicit Euler method specified by a first order BDF method. First, the full system is integrated one macro

step, then by using interpolated values for the slow system between t_n and t_{n+1}, the fast subsystem is integrated. This approach is chosen due to its stability using constant interpolation on $x_{\mathrm{S},n+1}$, and ease of implementation [5].

5 Verification and benchmark

The verification of the reduced order multirate method, utilising the discussed methods, and the netlist parser to create the network equations is done through numerical experiments. For experiment an academic circuit consisting of resistors, capacitors and diodes is considered. To benchmark the implementation of the reduced order multirate method it is compared against the original BDF method and a multirate BDF method.

5.1 Academic experiment

The academic circuit shown in Figure 1 is a combination of a short diode chain and then a long ladder of diodes and resistors. Similar to a standard diode chain model [10], this model contains sufficient redundancy to make it eligible for model order reduction. Furthermore, increasing the resistance for each R_i with $R < R_i < R_{i+1}$ makes that the ladder part of the circuits behaves on a slower timescale. This makes the circuit excellent for a time integration with the reduced order multirate approach.

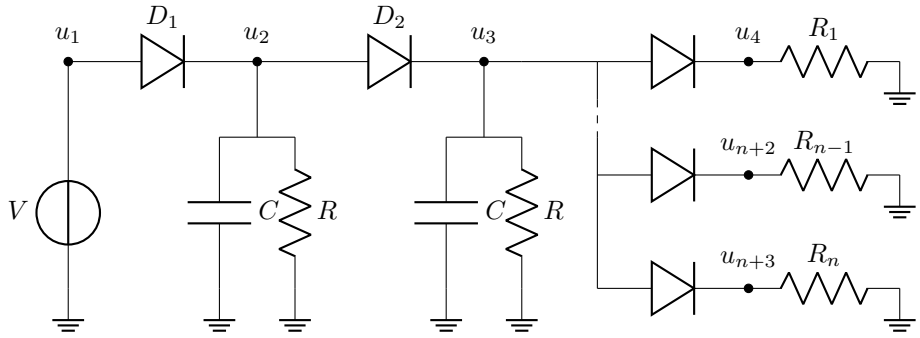

Figure 1: The academic diode chain test model with redundancy.

6 Implementation and Results

The reduced order multirate simulation of circuits program is implemented using MATLAB R2019b. The main file to run the benchmark is `ROMSOC.m`. This file can be run 'as is' to replicate the results of this paper. The file starts by including the sub-folders in which numerical methods and netlist parsing is encapsulated. For the reduced order multirate scheme benchmark the file

`multirate_example_long_nonlin.cir` is used as netlist file. This file is parsed and then the resulting equations are used for the circuit simulation. Should you desire to simulate other circuits, you can create your own netlist according to the syntax used in the example netlists. Do notice that in the current version the indices for the fast and slow subsystems must be partitioned manually. To create a longer diode chain for the academic diode chain test model, please use the `writeNetlist.m` file and change the length of the second for-loop.

In the main file, first a reference solution is obtained by an integration with the MATLAB integrator `ode15`. Then this reference solution is used to create the reduced order basis used in the model order reduction techniques. These bases and parameters are stored in the `mor_object` data structure and passed along each integrator.

To benchmark the convergence of the reduced order multirate method the simulation is run for an increasing number of time steps. The other simulation parameters are given in the box below.

Simulation parameters of the academic model		
Starting time	t_0	0 s
Ending time	t_N	0.004 s
Number of steps	N	[100 200 400 800]
Multirate factor	m	20
Newton tolerance	tol	10^{-8}
Original dimension	n	3000
Reduced dimension	r	2
Hyper-reduction factor	g	23
Voltage source	V	$5\sin(40 \cdot 2\pi t)$ V
Resistance	R	1000 Ω
Resistance	R_i	$i \cdot 1000$ Ω
Capacitance	C	10 μF
Diode saturation current	I_S	10^{-12} A

Regarding the convergence of the reduced order multirate integration scheme, the left figure in Figure 2 illustrates the order 1 convergence rate. We see that the reduced order multirate accuracy is nearly identical to that of the full order solutions. Furthermore, in the right figure of Figure 2 it shows that this accuracy is achieved with a significant reduction in computational time. The computational effort is almost a order of magnitude lower for the reduced schemes, while the precision is maintained. The positive effects of model order reduction, multirate time integration and the combination of both is evident.

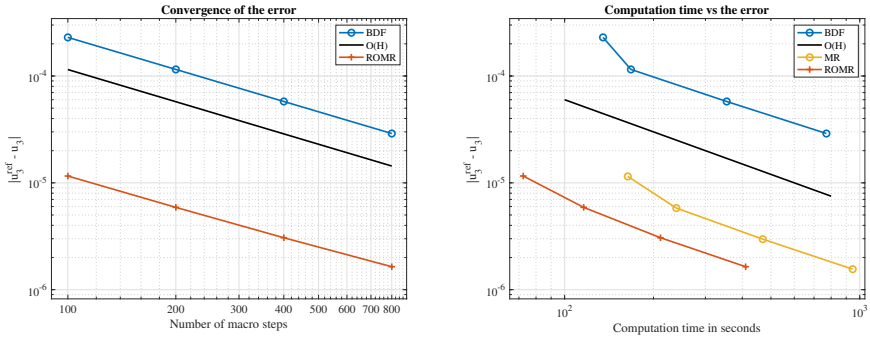

Figure 2: Convergence of the numerical schemes, where the error is plotted against the number of macro steps (left). Computational effort of the numerical schemes, where the error is plotted against the computation time in seconds (right). The error is defined as the absolute value between the computed voltage and reference voltage for the output node. The MR error is omitted in the left figure as the difference introduced by the reduction is negligible.

7 Conclusion

In this document a clear definition of a benchmark case for the reduced order multirate method is presented. Along with the source code distributed with this document a circuit simulation incorporating this new method can be run, and the result are easily verified. The combination of multirate time integration and model order reduction into one integration scheme shows a clear advantage for circuits with large redundancy. Higher order schemes are under development and will be incorporated in the final deliverable.

Acknowledgements

The authors are indebted to the funding given by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 765374, ROMSOC.

References

[1] M.W.F.M. Bannenberg, A. Ciccazzo, and M. Günther. Coupling of model order reduction and multirate techniques for coupled dynamical systems. *Applied Mathematics Letters*, 112:106780, 2021.

[2] MWFM Bannenberg, A Ciccazzo, and M Günther. Reduced order multirate schemes for coupled differential-algebraic systems. *Applied Numerical Mathematics*, 2021.

[3] M.W.F.M. Bannenberg, F. Kasolis, M. Günther, and M. Clemens. Maximum entropy snapshot sampling for reduced basis modelling. 2020.

[4] H. Broer and F. Takens. *Dynamical systems and chaos*. Springer-Verlag New York, 2011.

[5] C. Hachtel, A. Bartel, M. Günther, and A. Sandu. Multirate implicit Euler schemes for a class of differential–algebraic equations of index-1. *J. Comput. Appl. Math.*, page 112499, 2019.

[6] F. Kasolis, D. Zhang, and M. Clemens. Recurrent quantification analysis for model reduction of nonlinear transient electro-quasistatic field problems. In *International Conference on Electromagnetics in Advanced Applications (ICEAA 2019)*, pages 14–17, 2019.

[7] Howell A.M. Tong. *Dimension estimation and models*, volume 1. World Scientific, 1993.

[8] B.L. van der Waerden. *Mathematical Statistics*. Springer, 1969.

[9] A. Verhoeven. *Redundancy reduction of IC models : by multirate time-integration and model order reduction*. PhD thesis, Department of Mathematics and Computer Science, 2008.

[10] A. Verhoeven, J. Ter Maten, M. Striebel, and R. Mattheij. Model order reduction for nonlinear ic models. In *IFIP Conference on System Modeling and Optimization*, pages 476–491. Springer, 2007.

[11] G. Wanner and E. Hairer. *Solving ordinary differential equations II*, volume 375. Springer Berlin Heidelberg, 1996.