Some contributions to the model order reduction of large scale non-linear electric circuits

Juan Pablo Amorocho D.[1,*] and **Heike Faßbender**[1,**]

[1] Institut Computational Mathematics, TU Braunschweig, D-38023 Braunschweig

Model order Reduction (MOR) has become an ubiquitous technique in the simulation of large-scale dynamical systems (i.e. 10^4 and more equations). One technique for non-linear MOR is the trajectory piececewise-linear approach (TPWL) [1]. TPWL approximates a non-linear differential system by a weighted sum of linear systems which have a significantly reduced number of equations. One open question is which weight representations provide physical meaning of the weighted sum [2]. In this article we propose two representations.

1 The trajectory piecewise-linear approach (TPWL)

Here we apply the trajectory piecewise-linear approach to the non-linear dynamical system

$$\boldsymbol{x}'(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{D}\boldsymbol{v}(t),\ \boldsymbol{x}(0) = \boldsymbol{x}_0,\ 0 \le t \le T \tag{1}$$

where $\boldsymbol{x} : \mathbb{R} \longrightarrow \mathbb{R}^N$ is the function describing the state of the system, $\boldsymbol{f} : \mathbb{R}^N \longrightarrow \mathbb{R}^N$ a continuous non-linear function, $\boldsymbol{D} \in \mathbb{R}^{N\times M}$ the input-mapping, $\boldsymbol{v} : \mathbb{R} \longrightarrow \mathbb{R}^M$ the system's input. Furthermore the first order approximation of $\boldsymbol{f}$ around $(t_i, \boldsymbol{x}_i)$ is given by $\boldsymbol{f}_{\text{lin}}(\boldsymbol{x}) = \boldsymbol{f}_i + \boldsymbol{F}_i(\boldsymbol{x} - \boldsymbol{x}_i)$, where $\boldsymbol{F}_i = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}=\boldsymbol{x}_i}$ and $\boldsymbol{f}_i = \boldsymbol{f}(\boldsymbol{x}_i)$. Approximating $\boldsymbol{f}$ by $\boldsymbol{f}_{\text{lin}}$ in (1) yields the linear ODE $\boldsymbol{x}'_{\text{lin}}(t) = \boldsymbol{F}_i\boldsymbol{x}_{\text{lin}}(t) + \boldsymbol{B}_i\boldsymbol{u}(t)$, where $\boldsymbol{B}_i = [\boldsymbol{D}, \boldsymbol{f}_i - \boldsymbol{F}_i\boldsymbol{x}_i]$ and $\boldsymbol{u}(t) = [\boldsymbol{v}(t), 1]^{\mathsf{T}}$. The TPWL works as follows: First, one simulates (1) to extract s linear ODEs which approximate (1) at $(t_i, \boldsymbol{x}(t_i))$; the first one is always located at $(0, \boldsymbol{x}_0)$. Next, it constructs a projector $\boldsymbol{V} \in \mathbb{R}^{N\times q}$, $\boldsymbol{V}^{\mathsf{T}}\boldsymbol{V} = \boldsymbol{I}$, where $q \ll N$. Applying a Galerkin reduction yields the systems $\boldsymbol{z}'(t) = \hat{\boldsymbol{F}}_i\boldsymbol{z}(t) + \hat{\boldsymbol{B}}_i\boldsymbol{u}(t)$ where $\hat{\boldsymbol{F}}_i = \boldsymbol{V}^{\mathsf{T}}\boldsymbol{F}_i\boldsymbol{V}$, $\hat{\boldsymbol{B}}_i = \boldsymbol{V}^{\mathsf{T}}\boldsymbol{B}_i$, $\boldsymbol{x}_{\text{lin}}(t) \approx \boldsymbol{V}\boldsymbol{z}(t)$ and $\boldsymbol{z}(t) \in \mathbb{R}^q$. Finally, the TPWL model consists of the weight convex combination of these s reduced models:

$$\boldsymbol{z}'(t) = \sum_{i=0}^{s-1} w_i(\boldsymbol{z}(t))(\hat{\boldsymbol{F}}_i\boldsymbol{z}(t) + \hat{\boldsymbol{B}}_i\boldsymbol{u}(t)),\ \text{where} \sum_{i=0}^{s-1} w_i(\boldsymbol{z}) = 1 \text{ and } w_i(\boldsymbol{z}) \ge 0 \text{ for all } \boldsymbol{z}. \tag{2}$$

In [1], $w_i(\boldsymbol{z}(t)) = \frac{\exp(-\beta\cdot\|\boldsymbol{z}(t)-\boldsymbol{z}_i\|_2/\min_i(\|\boldsymbol{z}(t)-\boldsymbol{z}_i\|_2))}{\sum_{i=0}^{s-1}\exp(-\beta\cdot\|\boldsymbol{z}(t)-\boldsymbol{z}_i\|/\min_i(\|\boldsymbol{z}(t)-\boldsymbol{z}_i\|_2))}$ is suggested, where $\boldsymbol{x}_i = \boldsymbol{V}\boldsymbol{z}_i,\ i = 0, \dots, s-1$ and β is a positive scalar parameter - usually set to 25 - that makes the exponential function to decay fast. In [3–5], other weight representations involving the exponential function are proposed.

2 Proposed weighting functions

The first proposed weight function is $w_i(\boldsymbol{z}(t)) = 1$, if $\arg\min_i \|\boldsymbol{V}\boldsymbol{z}(t) - \boldsymbol{x}_i\|_2$; $w_i(\boldsymbol{z}(t)) = 0$ otherwise. We refer to it as the closest state weight function. Its behavior is very similar to the above weight function in [1] and its physical meaning is easy to grasp: At time t, let $\boldsymbol{x}_i$ be the closest state to $\boldsymbol{V}\boldsymbol{z}(t)$, then (2) approximates (1) just by the reduced linear ODE around $\boldsymbol{x}_i$.

Our second weight representation is a function of time only. At each t_i we define a piecewise weight function $w_i(t)$ as

$$w_i(t) = \begin{cases} w_i^l(t) & \text{if } t \in [t_{i-1}, t_i] \\ w_i^r(t) & \text{if } t \in [t_i, t_{i+1}] \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i = 1, \dots, s-2.$$

Here, the functions $w_i^l(t)$ and $w_i^r(t)$ are cubic polynomials such that $w_i^l(t_i) = w_i^r(t_i) = 1$, $w_i^l(t_{i-1}) = \frac{d}{dt}w_i^l(t_{i-1}) = \frac{d}{dt}w_i^l(t_i) = 0$ and $w_i^r(t_{i+1}) = \frac{d}{dt}w_i^r(t_i) = \frac{d}{dt}w_i^r(t_{i+1}) = 0$. For $i = 0$ and $i = s-1$ only 1 polynomial is needed in each case. If $t_{s-1} < T$, then we need an additional condition for w_{s-1}: $w_{s-1}(t) = 1, t \ge t_{s-1}$. We call this weight function cubic splines. The convexity conditions shown in (2) are now imposed in time. With cubic splines, (2) approximates (1) for each t by at most two linear ODEs.

---

[*] Corresponding author E-mail: `jamoroch@tu-braunschweig.de`
[**] E-mail: `h.fassbender@tu-braunschweig.de`

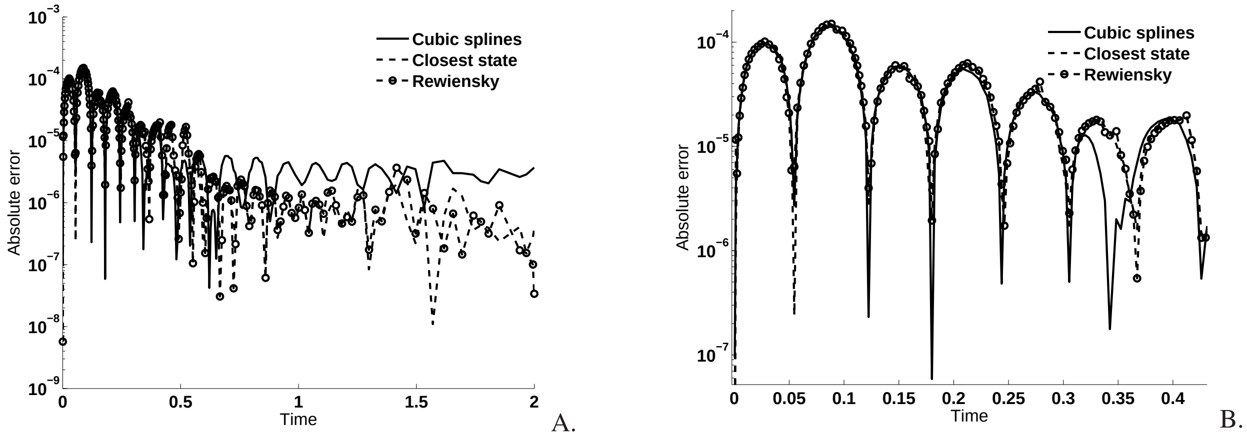

Fig. 1 Absolute error of $x_1(t)$.

Table 1 Relative error measure E_r

	Rewiensky	Closest state	Cubic splines
E_r	1.93×10^{-2}	1.91×10^{-2}	**1.83**$\times 10^{-2}$

3 Numerical experiments

We have tested our weight representations on a non-linear transmission line [1]

$$
\begin{aligned}
x_1'(t) &= -2x_1(t) + x_2(t) - i_{d_{1,0}} - i_{d_{1,2}} + v(t) \\
x_j'(t) &= x_{j-1}(t) - 2x_j(t) + x_{j+1}(t) + i_{d_{j-1,j}} - i_{d_{j,j+1}},\ j = 2, \ldots, N-1 \\
x_N'(t) &= x_{N-1} - x_N + i_{d_{N-1,N}},
\end{aligned}
$$

where $i_{d_{k,l}}(t) = \exp(40(x_k(t) - x_l(t))) - 1$ is the diode's current between nodes k and l. $l = 0$ means the ground node. Moreover the input $v(t)$ is $v(t) = \cos(16\pi t)e^{-2\pi t}, N = 500$ and $0 \le t \le 2$. In the first step of TPWL 48 linear models were generated with the algorithm of [1]. The applied linear MOR is the algorithm PRIMA [6], reducing only the linear model at the initial condition. PRIMA yields the projector $\boldsymbol{V}$ of rank 50. Figure 1 shows the absolute error $E_{abs}(t) = \|x_1^{exact}(t) - x_1^{tpwl}(t)\|_2$, Table 1 reports the relative error measure $E_r = \max_t E_{abs}(t)/\max_t \|x_1^{exact}(t)\|_2$. The plot shows that the closest state weight function performs similar to the original Rewiensky's ansatz; at the same time the cubic splines does not, on average, underperfom. Figure B shows a detail of figure A in the time interval $0 \le t < 0.5$. Here, where the solution starkly oscillates, all three weight representations deliver the same absolute error. The relative error for all three weight functions is comparable. The experiments were done on a Intel Xeon CPU 3.66 Ghz machine running a Linux kernel 2.6.16.27 using Matlab R2008b. The function `ode23s` - without additional options - has been used to extract the linear models and to simulate the TPWL model.

Acknowledgements This work, part of the SyreNe network, is supported by the German Federal Ministry of Education and Research (BMBF) grant no. 03FAPAE2.

References

[1] M. Rewiensky, A trajectory piecewise-linear Approach to Model Order Reduction of Nonlinear Dynamical Systems, PhD thesis, Massachusetts Institute of Technology, 2003.

[2] M. Striebel and J. Rommes, Model order reduction of non-linear systems: status, open issues, and applications, Tech. Rep. CSC/08-07, Technische Universität Chemnitz, 2008. Germany

[3] T. Voß, Model reduction for nonlinear differential algebraic equations, Master's thesis, Universität Wuppertal, 2005. Germany

[4] N. Dong and J. Roychowdhury, Automated nonlinear macromodelling of output buffers for high-speed digital applications, in: DAC '05: Proceedings of the 42nd annual Design Automation Conference, (ACM, New York, NY, USA, 2005), pp. 51–56.

[5] S. K. Tiwary and R. A. Rutenbar, Scalable trajectory methods for on-demand analog macromodel extraction, in: DAC '05: Proceedings of the 42nd annual Design Automation Conference, (ACM, New York, NY, USA, 2005), pp. 403–408.

[6] A. Odabasioglu, M. Celik, and L. T. Pileggi, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **17**(8), 645–654 (1998).