Convergence Properties of Hierarchical Co-simulation Approaches

Irene Hafner[1(✉)] and Niki Popper[1,2]

[1] dwh GmbH, Neustiftgasse 57-59, 1070 Vienna, Austria
{irene.hafner,niki.popper}@dwh.at
[2] Institute of Information Systems Engineering, TU Wien, Favoritenstraße 9-11, 1040 Vienna, Austria

Abstract. In this paper, local error estimates for hierarchical co-simulation approaches are presented. In hierarchical structures, systems with stronger dependencies on one another, which frequently occur in large-scale cyber-physical systems, may be combined in further co-simulations on one or more lower levels. This allows the selection of individual synchronization times for subsystems in these co-simulations. The estimates presented in this paper show that with this approach, no additional errors compared to traditional co-simulation are to be expected: on the contrary, results from the simulation of a benchmark example show that in case of sensible selection of lower-level couplings, accuracy and stability may even be increased as error propagation slows down.

Keywords: Co-simulation · Hierarchy · Consistency

1 Introduction

Hierarchical structures are no novelty in modeling and simulation in general, confer for example the Discrete Event System Specification (DEVS [19]) or partitioned integration methods [6,7,12,14]. However, hierarchical co-simulation as explained in the following is scarcely found in the literature. Although several frameworks and standards do not prohibit further co-simulations within a co-simulation, and some authors acknowledge the possibility of nested co-simulation [16,17], hierarchical co-simulation has, to the best of our knowledge, not been investigated with regard to error estimates up to now. Compared to hierarchical partitioned multirate schemes, subsystems may still be implemented in individually suitable simulation tools in a hierarchical co-simulation approach.

The idea of the introduction of further co-simulation levels is illustrated in Fig. 1. Such further division and nesting of co-simulations can be motivated by highly diverse time constants or other subsystem properties that require closer interaction between certain subsystems. In a traditional co-simulation approach, this could enforce a rather small macro step and thus, synchronization of all subsystems and consequently high computation time. With the introduction of further levels, more closely dependent subsystems may communicate with a small

A. Cerone et al. (Eds.): SEFM 2021 Workshops, LNCS 13230, pp. 156–172, 2022.
https://doi.org/10.1007/978-3-031-12429-7_12

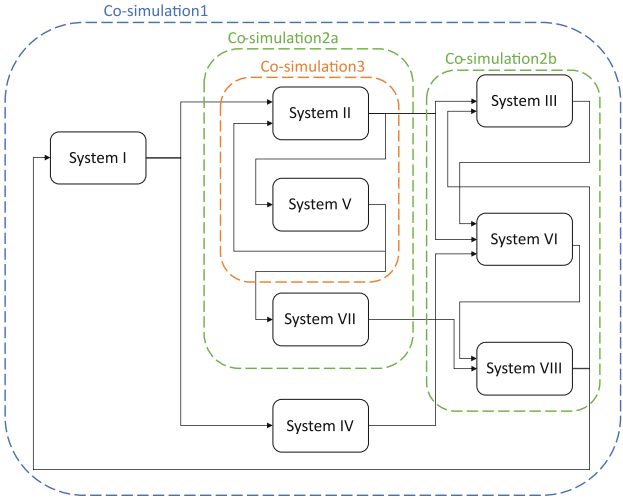

Fig. 1. Schematic depiction of a hierarchical co-simulation approach. Coordination takes place on several levels by one top-level co-simulation that manages the communication between subsystems and further co-simulations. These may again coordinate subsystems and co-simulations on lower levels [9].

macro step on a lower-level co-simulation while exchanging values with all other systems on a distinctly larger upper-level macro step, thus increasing accuracy without drastically slowing down the whole simulation.

A typical application example with these properties would be a manufacturing process where machines have to exchange data rather frequently with logistics while only from time to time transferring their waste heat data to a slow varying, thermal room model. This, in turn, has to be synchronized with an HVAC simulation controlling the room temperature. The latter would not require any communication with machines or logistic devices themselves, let alone evaluation and data exchange at the same, considerably small, time steps. Holistic simulation of urban energy systems likewise intrinsically brings along several different levels of consideration: households, factories, traffic, network, and power plants can each prove complex enough to be addressed by individual co-simulations, which then have to communicate in order to portray the overall system.

In the following, investigations on convergence of the proposed method are presented, starting with estimates on the consistency error for traditional, single-level co-simulation, extending them to hierarchically structured approaches and presenting error studies that illustrate the improvement in accuracy. Investigations on zero-stability and numerical stability, which are essential in addition to consistency to guarantee convergence in case of coupled DAE systems or ODEs with multi-step integration algorithms, are found in [8,9]. There it is shown that stability issues can be tackled by introducing another layer of communication

instead of having to decrease the overall communication step size, thus providing an innovative method for stabilization.

2 Consistency

It has been shown in the literature that local error control is a valid method to bound the global co-simulation error (see f.i. [1,4,20]). This justifies investigating the consistency error, i.e. the error of the method in one step, in a co-simulation. For this aim we need to start by calling to mind some background information on numerics of differential equations.

In the following, we consider a uniquely solvable ordinary differential equation initial value problem

$$\dot{\boldsymbol{x}} = f(t, \boldsymbol{x}), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{1}$$

with Lipschitz continuous right side f with respect to $\boldsymbol{x}$.

For a given approximation $\boldsymbol{x}_{t_n+h}$ of $\boldsymbol{x}$ at time t_n+h by a numerical integration method with step size h, the consistency error is defined as the error of the method in one step and therefore, calculated by

$$\mathcal{E}(t_n, \boldsymbol{x}_n, h) = \boldsymbol{x}(t_n + h) - \boldsymbol{x}_{t_n+h} \tag{2}$$

for given initial values $\boldsymbol{x}(t_n) = \boldsymbol{x}_n$. A method is called consistent if

$$\lim_{h \to 0} \left(\frac{\mathcal{E}(t_n, \boldsymbol{x}_n, h)}{h} \right) = \mathbf{0} \tag{3}$$

for every choice of $t_n, \boldsymbol{x}_n$. A method is called consistent of order p if there exists a constant $C > 0$ with

$$\left\| \frac{\mathcal{E}(t_n, \boldsymbol{x}_n, h)}{h} \right\| \leq C \cdot h^p. \tag{4}$$

Since the consistency error is a measure for the *local* error of a method, state values are taken to be exact for all previous points in time.

Remark 1. In case they are not directly needed in the following calculations, the initial values $t_n, \boldsymbol{x}_n$ will be omitted in the notation of $\mathcal{E}$ to simplify the notation.

Important for the error estimates following below are Gronwall's Lemma (Theorem 1) and "the fundamental lemma" (Theorem 2).

Theorem 1. (Gronwall's Lemma *[15]*). *Let the real function $m(t)$ be continuous in $J := [0, a]$, and let*

$$m(t) \leq \alpha + \beta \int_0^t m(\tau) d\tau \quad \text{in } J \text{ with } \beta > 0$$

then

$$m(t) \leq \alpha e^{\beta t} \quad \text{in } J.$$

Theorem 2. (The "fundamental lemma" [10]). *Supposing that* $\boldsymbol{x}(t)$ *is a solution of the system of differential Eqs. 1 with* f *Lipschitz continuous in the second argument with Lipschitz constant* L*, and* $\boldsymbol{v}(t)$ *an approximate solution fulfilling*

$$\|\dot{\boldsymbol{v}}(t) - f(t, \boldsymbol{v}(t))\| \leq \epsilon,$$

then, for $t \geq t_0$*, we have the error estimate*

$$\|\boldsymbol{x}(t) - \boldsymbol{v}(t)\| \leq \|\boldsymbol{x}(t_0) - \boldsymbol{v}(t_0)\| \, e^{L(t-t_0)} + \frac{\epsilon}{L}\left(e^{L(t-t_0)} - 1\right).$$

Remark 2. If $\boldsymbol{v}$ is also an exact solution of $\dot{\boldsymbol{x}} = f(t, \boldsymbol{x})$, from Theorem 2 follows

$$\|\boldsymbol{x}(t) - \boldsymbol{v}(t)\| \leq \|\boldsymbol{x}(t_0) - \boldsymbol{v}(t_0)\| \, e^{L(t-t_0)},$$

which directly implies that in case of the same initial values, $\boldsymbol{v}$ is identical to $\boldsymbol{x}$.

2.1 Consistency in Co-simulation

To investigate consistency in co-simulation, we consider a system of N coupled ODEs given as follows[1]:

$$\dot{\boldsymbol{x}}^i(t) = \boldsymbol{f}^i(\boldsymbol{x}^i, \boldsymbol{u}^i, t), \quad \boldsymbol{x}^i(t_0) = \boldsymbol{x}_0^i \tag{5a}$$

with $i = I, \ldots, N$, $\boldsymbol{x}^i \in \mathbb{R}^{n_x^i}$, $\boldsymbol{u}^i \in \mathbb{R}^{n_u^i}$, and

$$\boldsymbol{u}^i = \boldsymbol{L}^i \boldsymbol{x} = \begin{bmatrix} \boldsymbol{L}^{i,I} & \ldots & \boldsymbol{L}^{i,i-1} & 0 & \boldsymbol{L}^{i,i+1} & \ldots & \boldsymbol{L}^{i,N} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^I \\ \vdots \\ \boldsymbol{x}^{i-1} \\ \boldsymbol{x}^i \\ \boldsymbol{x}^{i+1} \\ \vdots \\ \boldsymbol{x}^N \end{bmatrix} \tag{5b}$$

with $\boldsymbol{L}^{i,j} \in \mathbb{R}^{n_u^i \times n_x^j} \quad \forall i, j \in \{I, \ldots, N\}$ and the elements of $\boldsymbol{L}^{i,j}$ being equal to zero or one, thus describing the output-input dependencies between the individual subsystems. Thereby, we assume again a unique solution and Lipschitz continuous right-side functions $\boldsymbol{f}^i$ in the first and second argument.

In the following, investigations on convergence of traditional co-simulation analogously as given by Knorr [11][2] are presented and extended on hierarchical

[1] Notation with elements of $\mathbb{G} := \{I, II, \ldots\}$ is used to avoid confusion with exponents and allow easy identification of subsystems. In arithmetic operations where elements of $\mathbb{G}$ and $\mathbb{N}$ are mingled, these are to be understood as operations between elements of $\mathbb{N}$ by assigning every element of $\mathbb{G}$ its image under the bijection that uniquely assigns the i-th element of $\mathbb{G}$ the i-th element of $\mathbb{N}$.

[2] The investigations in [11] are restricted to two participating subsystems where the larger micro step size is also taken as macro step size. Following this strategy, we allow an arbitrary number of participating subsystems and macro step size H with the possibility of $H > h_i$ for all subsystem solver step sizes h_i in this work.

approaches in Sect. 2.2. We start by considering the i-th subsystem of (5). In case of a multirate co-simulation, values $\boldsymbol{u}^i$ have to be extrapolated in between two synchronization time steps and will be named $\tilde{\boldsymbol{u}}^i$. Depending on the order q_i of the chosen extrapolation method, $\left\|\boldsymbol{u}^i(t_n+h)-\tilde{\boldsymbol{u}}^i(t_n+h)\right\| \leq Ch^{q_i+1}$ with a constant $C>0$ holds for a step of size $h>0$ assuming $\tilde{\boldsymbol{u}}^i(t_n)=\boldsymbol{u}^i(t_n)$. Further, $\boldsymbol{x}^i(t)$ will denote the exact solution of (5a) and $\tilde{\boldsymbol{x}}^i(t)$ the exact solution of

$$\dot{\boldsymbol{x}}^i(t)=\boldsymbol{f}^i(\boldsymbol{x}^i,\tilde{\boldsymbol{u}}^i,t), \quad \boldsymbol{x}^i(t_0)=\boldsymbol{x}_0^i. \tag{6}$$

The approximated solution of (6) at $t_{n,k}$ will be named $\tilde{\boldsymbol{x}}_{n,k}$.

To begin with, we regard the error $\mathcal{E}^i(t_{n,k},\boldsymbol{x}_{n,k},h_i)$ of the i-th subsystem in one micro step h_i at $t_{n,k}$, where n is the current macro step and k the current micro step, counted anew for each macro interval. Thus $t_{n+1}:=t_{n+1,0}=t_{n,m_i}=t_n+m_i\cdot h_i=t_n+H$ in case of m_i micro steps per macro step, hence m_i denoting the multirate factor of subsystem i in case of fixed, equidistant micro steps which are integer divisors of the (also fixed) macro step size H, which we will assume w.l.o.g.[3] in the following calculations.

Starting with the consistency of the integration of every subsystem for one micro step, we will deduce consistency of the integration of every subsystem for one macro step and further of the co-simulation.

Lemma 1 (Consistency error for one micro step). *Let p_i denote the consistency order of the original method and q_i the order of extrapolation for input values $\boldsymbol{u}^i$. Then*

$$\left\|\frac{\mathcal{E}^i(t_{n,k},\boldsymbol{x}_{n,k},h_i)}{h_i}\right\| = \mathcal{O}\left(h_i^{\min\{p_i,q_i+1\}}\right). \tag{7}$$

Proof. Considering exact values at $t_{n,k}$, per definition

$$\left\|\mathcal{E}^i(t_{n,k},\boldsymbol{x}_{n,k},h_i)\right\| = \left\|\boldsymbol{x}^i(t_{n,k}+h_i)-\tilde{\boldsymbol{x}}^i_{n,k+1}\right\| = \left\|\boldsymbol{x}^i(t_{n,k+1})-\tilde{\boldsymbol{x}}^i_{n,k+1}\right\| \tag{8}$$

with the notation described above. Adding and subtracting $\tilde{\boldsymbol{x}}(t_{n,k+1})$ gives

$$\left\|\mathcal{E}^i(t_{n,k},\boldsymbol{x}_{n,k},h_i)\right\| \overset{\text{triangle inequ.}}{\leq} \left\|\boldsymbol{x}^i(t_{n,k+1})-\tilde{\boldsymbol{x}}^i(t_{n,k+1})\right\| + \underbrace{\left\|\tilde{\boldsymbol{x}}^i(t_{n,k+1})-\tilde{\boldsymbol{x}}^i_{n,k+1}\right\|}_{\leq C_{i,1}\cdot h_i^{p_i+1}}. \tag{9}$$

The second term of (9) is the difference of the exact to the approximated solution of the modified system (6) and is therefore bounded by $C_{i,1}\cdot h_i^{p_i+1}$ for a constant $C_{i,1}>0$ and with p_i being the order of the numerical integration method given for system i.

[3] All considerations can be performed analogously for unequally distanced grids with h_i taken as upper bound of all h_{i_j} with $i_j\in\{1,\ldots,m_{i_n}\}$ and m_{i_n} the number of micro steps of subsystem i in the n-th macro step. However, as this would only lead to more complex notation, we will restrict the step sizes as described above for reasons of clarity.

To provide an estimate for the first term in (9), we use the assumption that $\boldsymbol{x}(t)$ and $\tilde{\boldsymbol{x}}(t)$ are the exact solutions of (5a) and (6), respectively, and can therefore be replaced by the integral over their derivatives (since they fulfill conditions like uniqueness, continuity, and differentiability by definition):

$$\begin{aligned}\left\|\boldsymbol{x}^i(t_{n,k+1}) - \tilde{\boldsymbol{x}}(t_{n,k+1})\right\| &= \left\| \int_{t_{n,k}}^{t_{n,k+1}} \left(f^i(\boldsymbol{x}^i, \boldsymbol{u}^i, \tau) - f(\tilde{\boldsymbol{x}}^i, \tilde{\boldsymbol{u}}^i, \tau) \right) d\tau \right\| \\ &\leq \int_{t_{n,k}}^{t_{n,k+1}} \left\| f^i(\boldsymbol{x}^i, \boldsymbol{u}^i, \tau) - f(\tilde{\boldsymbol{x}}^i, \tilde{\boldsymbol{u}}^i, \tau) \right\| d\tau\end{aligned} \tag{10}$$

Adding and subtracting $f(\tilde{\boldsymbol{x}}^i, \boldsymbol{u}^i, \tau)$ gives with the triangle inequality

$$(10) \leq \int_{t_{n,k}}^{t_{n,k+1}} \left\| f^i(\boldsymbol{x}^i, \boldsymbol{u}^i, \tau) - f(\tilde{\boldsymbol{x}}^i, \boldsymbol{u}^i, \tau) \right\| d\tau + \int_{t_{n,k}}^{t_{n,k+1}} \left\| f^i(\tilde{\boldsymbol{x}}^i, \boldsymbol{u}^i, \tau) - f(\tilde{\boldsymbol{x}}^i, \tilde{\boldsymbol{u}}^i, \tau) \right\| d\tau$$

$$\overset{\text{Lipschitz}}{\leq} \int_{t_{n,k}}^{t_{n,k+1}} L_{f^i,x} \left\| \boldsymbol{x}^i - \tilde{\boldsymbol{x}}^i \right\| d\tau + \underbrace{\int_{t_{n,k}}^{t_{n,k+1}} L_{f^i,u} \underbrace{\left\| \boldsymbol{u}^i - \tilde{\boldsymbol{u}}^i \right\|}_{\leq C_{i,2} \cdot h_i^{q_i+1}} d\tau}_{\leq L_{f^i,u} \cdot C_{i,2} \cdot h_i^{q_i+2}} \tag{11}$$

with Lipschitz constants $L_{f^i,x}$ and $L_{f^i,u}$ of f^i with respect to $\boldsymbol{x}$ and $\boldsymbol{u}$, respectively, and q_i denoting the order of the extrapolation method for the approximation of $\tilde{\boldsymbol{u}}^i$. Declaring $C_{i,3} := L_{f^i,u} \cdot C_{i,2}$ and $\boldsymbol{m}(t) := \left\| \boldsymbol{x}^i(t) - \tilde{\boldsymbol{x}}^i(t) \right\|$, above estimates can be summarized as

$$\boldsymbol{m}(t_{n,k+1}) \leq \int_{t_{n,k}}^{t_{n,k+1}} L_{f^i,x} \left\| \boldsymbol{x}^i - \tilde{\boldsymbol{x}}^i \right\| d\tau + C_{i,3} \cdot h_i^{q_i+2}. \tag{12}$$

Now we can apply the Lemma of Gronwall (Theorem 1) to $\boldsymbol{m}$ with $\alpha = C_{i,3} \cdot h_i^{q_i+2}$ and $\beta = L_{f^i,x}$ and obtain

$$\boldsymbol{m}(t_{n,k+1}) \leq C_{i,3} \cdot h_i^{q_i+2} \cdot \underbrace{e^{L_{f^i,x} \cdot \overbrace{(t_{n,k+1} - t_{n,k})}^{h_i}}}_{= \sum_{j=0}^{\infty} \frac{(L_{f^i,x} \cdot h_i)^j}{j!}} = \mathcal{O}\left(h_i^{q_i+2} \right) \tag{13}$$

and therefore

$$\left\| \frac{\mathcal{E}^i(t_{n,k}, \boldsymbol{x}_{n,k}, h_i)}{h_i} \right\| \overset{(9),(13)}{\leq} C_{i,1} \cdot h_i^{p_i} + \mathcal{O}\left(h_i^{q_i+1} \right) = \mathcal{O}\left(h_i^{\min\{p_i, q_i+1\}} \right). \tag{14}$$

□

This shows that while consistency is maintained in co-simulation, the order may be reduced if the extrapolation order is chosen too low. Constant extrapolation, for example, only maintains the order of integration methods of order one. For higher-order methods, the order is reduced but the method remains consistent (as $\left\| \frac{\mathcal{E}^i(t_{n,k}, \boldsymbol{x}_{n,k}, h_i)}{h_i} \right\|$ still converges to zero, but only linearly). However, higher order extrapolation can also lead to increased stability issues, which is shown for example in [2].

Lemma 2 (Consistency error per subsystem for one macro step). *With the notations above*

$$\left\| \frac{\mathcal{E}^i(t_n, \boldsymbol{x}_n, H)}{H} \right\| = \mathcal{O}\left(H^{\min\{p_i, q_i+1\}} \right). \tag{15}$$

Proof. To extend the considerations for one micro step to one macro step, we will employ the method of "Lady Windermere's Fan", which is shown f.i. in [10,11]. The main idea of this approach is to describe the error of the approximate solution after an interval – in our case, a macro step – by the analytical solutions at every point of a refined mesh – in our case, every micro step – assuming an exact value at the beginning of the considered interval. This is illustrated for a one-dimensional problem in Fig. 2.

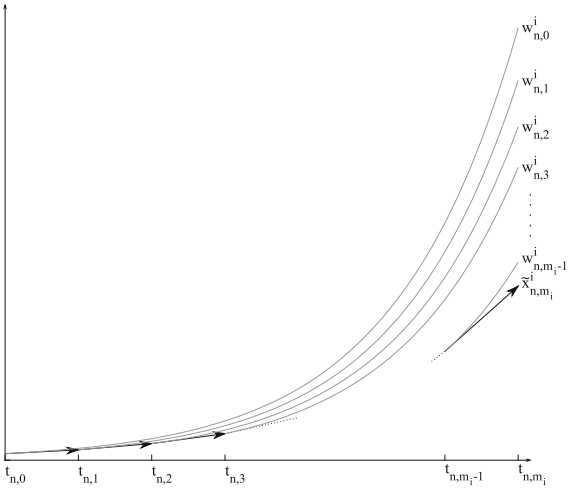

Fig. 2. "Lady Windermere's Fan": exact solutions at every time step of the numerical integration algorithm are used to describe the error of the approximate solution in one macro step (after [11]).

Let $\boldsymbol{w}^i_{n,k}(t), k = 0, \dots m_i$ denote the exact solution of system (6) but for the initial values $\boldsymbol{w}^i_{n,k}(t_{n,k}) = \tilde{\boldsymbol{x}}_{n,k}$, implying $\boldsymbol{w}^i_{n,0}(t) = \boldsymbol{x}^i(t)\ \forall t > t_n$ since we assume exact values at $t_{n,0}$. Then we can write

$$\left\|\mathcal{E}^i(t_n, \boldsymbol{x}_n, H)\right\| = \left\|\boldsymbol{x}^i(t_{n+1}) - \tilde{\boldsymbol{x}}^i_{n+1}\right\| = \left\|\boldsymbol{x}^i(t_{n,m_i}) - \tilde{\boldsymbol{x}}^i_{n,m_i}\right\| \tag{16}$$

$$\leq \sum_{k=0}^{m_i-1} \left\|\boldsymbol{w}^i_{n,k}(t_{n,m_i}) - \boldsymbol{w}^i_{n,k+1}(t_{n,m_i})\right\|. \tag{17}$$

Since $\boldsymbol{w}^i_{n,k}$ are solutions to the same system with different initial values, we can apply Theorem 2 know that every summand of (17) is bounded by

$$\left\|\boldsymbol{w}^i_{n,k}(t_{n,k+1}) - \boldsymbol{w}^i_{n,k+1}(t_{n,k+1})\right\| \cdot e^{L_{f^i,x} \cdot \overbrace{(t_{n,m_i} - t_{n,k+1})}^{(m_i-k-1)h_i}} \tag{18}$$

$$\Rightarrow \left\|\mathcal{E}^i(t_n, \boldsymbol{x}_n, H)\right\| \leq \sum_{k=0}^{m_i-1} \left\|\boldsymbol{w}^i_{n,k}(t_{n,k+1}) - \boldsymbol{w}^i_{n,k+1}(t_{n,k+1})\right\| \cdot e^{L_{f^i,x} \cdot (m_i-k-1)h_i}$$

$$\overset{\boldsymbol{w}^i_{n,k}(t_{n,k}) = \tilde{\boldsymbol{x}}_{n,k}}{=} \sum_{k=0}^{m_i-1} \left\|\boldsymbol{w}^i_{n,k}(t_{n,k+1}) - \tilde{\boldsymbol{x}}^i_{n,k+1}\right\| \cdot e^{L_{f^i,x} \cdot (m_i-k-1)h_i}.$$

As $\left\|\boldsymbol{w}^i_{n,k}(t_{n,k+1}) - \boldsymbol{x}^i_{n,k+1}\right\|$ is the error in one micro step, according to Lemma 1 we can estimate this term with $\mathcal{O}(h_i^{\min\{p_i+1,q_i+2\}})$. Therefore

$$\left\|\mathcal{E}^i(t_n, \boldsymbol{x}_n, H)\right\| \leq \mathcal{O}\left(h_i^{\min\{p_i+1,q_i+2\}}\right) \sum_{k=0}^{m_i-1} e^{L_{f^i,x} \cdot (m_i-k-1)h_i} \tag{19}$$

$$\leq \mathcal{O}\left(h_i^{\min\{p_i+1,q_i+2\}}\right) \cdot m_i \cdot e^{L_{f^i,x} \cdot (m_i-1)h_i} \tag{20}$$

$$\overset{h_i = H/m_i}{=} \mathcal{O}\left(\left(\frac{H}{m_i}\right)^{\min\{p_i+1,q_i+2\}}\right) \cdot m_i \cdot e^{L_{f^i,x} \cdot \frac{m_i-1}{m_i} H} \tag{21}$$

$$= \mathcal{O}\left(H^{\min\{p_i+1,q_i+2\}}\right) \tag{22}$$

$$\Rightarrow (15). \qquad \square$$

Corollary 1 (Consistency error of co-simulation). *With the notations above, consistency of the co-simulation in one macro step can be determined by*

$$\left\|\frac{\mathcal{E}(t_n, \boldsymbol{x}_n, H)}{H}\right\| = \mathcal{O}\left(H^{\min\limits_{i=I,\dots,N}\{p_i,q_i+1\}}\right), \tag{23}$$

whereby

$$\min_{i=I,\dots,N}\{p_i, q_i+1\} := \min_{i=I,\dots,N}\{\min\{p_i, q_i+1\}\} = \min\{\min_{i=I,\dots,N}\{p_i\}, \min_{i=I,\dots,N}\{q_i+1\}\}.$$

Proof. Since $\boldsymbol{x}(t)$ is given as concatenation of all $\boldsymbol{x}^i(t)$, $i = I, \dots, N$, the approximation of the overall system at a synchronization point t_{n+1} corresponds to the concatenation of the approximations of the states of the N individual subsystems. With this, we can simply infer

$$\|\mathcal{E}(t_n, \boldsymbol{x}_n, H)\| = \|\boldsymbol{x}(t_{n+1}) - \tilde{\boldsymbol{x}}_{n+1}\| \leq \sum_{i=0}^{N} \left\|\boldsymbol{x}^i(t_{n+1}) - \tilde{\boldsymbol{x}}^i_{n+1}\right\| \tag{24}$$

$$\leq N \cdot \mathcal{O}\left(H^{\min\limits_{i=1\dots N}\{p_i+1, q_i+2\}}\right) = \mathcal{O}\left(H^{\min\limits_{i=1\dots N}\{p_i+1, q_i+2\}}\right) \tag{25}$$

with the estimates from Lemma 2. (23) follows directly by division by H. □

These estimates show that while overall consistency is maintained in the co-simulation of ODE systems, the convergence order may be reduced in case of lower-order extrapolation of input values. Higher order extrapolation, while enhancing the order of consistency of the coupled method (bounded by the order of the original integration method), can also lead to increased stability issues, as shown f.i. in [1,2]. For DAEs that are only coupled via differential variables, the implicit function theorem (see e.g. [18]) implies that locally, an equivalent ODE system can be found for which above considerations also apply. In case of coupling via algebraic variables, similar estimates (in the sense of dependence on extrapolation orders) are given e.g. in [3,5].

2.2 Consistency in Hierarchical Co-simulation

Now we want to extend above investigations to co-simulation on several levels of hierarchy. From the estimates for traditional co-simulation, which only depend on the error introduced by extrapolation of external input values, we can already expect that this property is not affected by the method used in the respective other subsystems or the time steps and further synchronizations happening there in-between. For detailed estimation, we will first consider the simplest case where hierarchical co-simulation can be applied: Three subsystems of which w.l.o.g. Systems II and III are co-simulated on the lowest level and this co-simulation communicates again on the topmost level with the simulation of System I, as illustrated in Fig. 3.

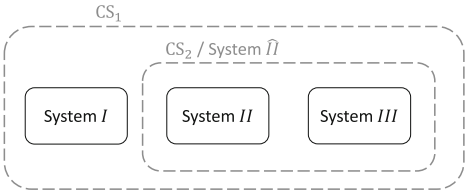

Fig. 3. Illustration of hierarchical co-simulation of three systems on two levels. Co-simulation CS_1 coordinates System I and System $\widehat{II}$, i.e. co-simulation CS_2, which manages the communication between systems II and III.

The co-simulation between Systems II and III will be called CS_2 henceforth, and the corresponding system seen from the perspective of the upper level System $\widehat{II}$. The top-level co-simulation (CS_1) macro step will be denoted H_1

and the second-level co-simulation macro step H_2. For CS_1, we start with the error in one macro step H_1 of System I, for which we obtain from Lemma 2

$$\left\| \frac{\mathcal{E}^I(H_1)}{H_1} \right\| = \mathcal{O}\left(H_1^{\min\{p_I, q_I+1\}} \right). \tag{26}$$

For System $\widehat{II}$, we start by applying Corollary 1 to CS_2, which yields for one step of size H_2

$$\left\| \frac{\mathcal{E}^{\widehat{II}}(H_2)}{H_2} \right\| = \mathcal{O}\left(H_2^{\min\limits_{i=II,III}\{p_i, q_i+1\}} \right). \tag{27}$$

To estimate the error in one macro step H_1, we can repeat the strategy from the proof of Lemma 2 with M_2 describing the quotient of H_1 and H_2 and obtain

$$\left\| \frac{\mathcal{E}^{\widehat{II}}(H_1)}{H_1} \right\| = \mathcal{O}\left(H_1^{\min\limits_{i=II,III}\{p_i, q_i+1\}} \right) \tag{28}$$

and further for the top-level co-simulation CS_1 with (26), (28) and Corollary 1

$$\left\| \frac{\mathcal{E}(H_1)}{H_1} \right\| = \mathcal{O}\left(H_1^{\min\limits_{i=I,II,III}\{p_i, q_i+1\}} \right) \tag{29}$$

and therefore consistency. The order again depends on the extrapolation and consistency orders of all subsystems. This can also be concluded for arbitrary levels of hierarchy and participating subsystems, as Theorem 3 shows.

Theorem 3 (Consistency error of hierarchical co-simulation). *In a hierarchical co-simulation with a total of N participating subsystems, consistency orders p_i, $i = I, \ldots, N$ of their corresponding integration algorithms and extrapolation orders q_i, $i = I, \ldots, N$, the consistency error of the overall co-simulation with macro step H can be estimated as*

$$\left\| \frac{\mathcal{E}(H)}{H} \right\| = \mathcal{O}\left(H^{\min\limits_{i=I,\ldots,N}\{p_i, q_i+1\}} \right). \tag{30}$$

Proof. To begin with, we need to establish comprehensible notation of all considered systems, co-simulations, and step sizes. For this purpose, all participating simulations are depicted in a tree structure, see Fig. 4. We will start from the topmost level, naming the overall co-simulation $S_{1,1}$. Beneath $S_{1,1}$, all further simulations unfold on J levels in total. On every level $j \in 1, \ldots, J+1$ all simulations – be they co-simulations themselves or "leaf" nodes without further branching beneath – are numbered from 1 to K_j. This means that on level j, we find simulations $S_{j,k}$ with $k = 1 \ldots K_j$. While the ordering of these may be arbitrary, this notation is necessary to uniquely identify every co-simulation on every level in a fairly intelligible notation. Nevertheless, to clarify the belonging to the respective co-simulation, the sub-simulations of one node, i.e. all $N_{j,k}$ simulations coordinated by one co-simulation $S_{j,k}$ may be identified by $S^I_{j,k}, S^{II}_{j,k}, \ldots, S^{N_{j,k}}_{j,k}$

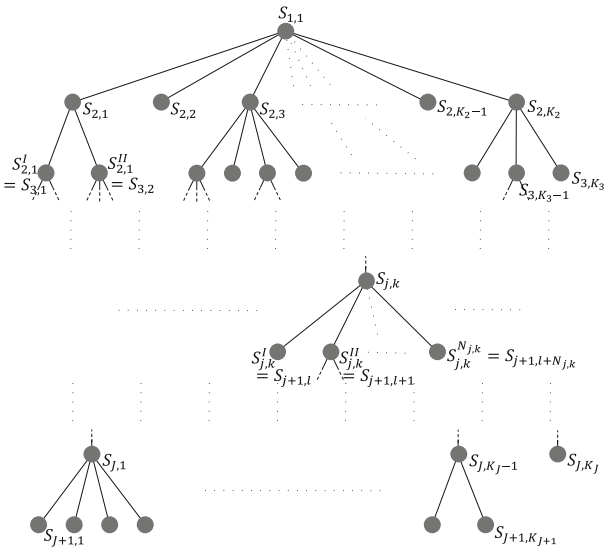

Fig. 4. Illustration of the co-simulation hierarchy in a tree structure.

in addition. This means that the $i-$th subsimulation of $S_{j,k}$ may be called $S^i_{j,k}$ and equals, using the notation on the next level, $S_{j+1,l}$ for one $l \in \{1, \ldots, K_{j+1}\}$:

$$S^i_{j,k} = S_{j+1,l} \quad \text{for } l = i + \sum_{m=1}^{k-1} N_{j,m} \tag{31}$$

Note that naturally, the sum of all simulations that are co-simulated by simulations on level j equals the number of simulations on level $j+1$ with the convention that for leaf nodes, $N_{j,k} := 0$.

In analogy to above example with three systems co-simulated on two levels, (30) follows from Lemmata 1, 2 and Corollary 1 when approached bottom-up with induction: On the deepest level $J+1$, we only have leaf nodes. These systems $S_{J+1,l}, l = 1, \ldots, K_{J+1}$ are integrated with their individual time step $h_{J+1,l}$ and are coordinated by a co-simulation on level J. By considering one of these co-simulations $S_{J,k}$ with macro step size $H_{J,k}$ and its sub-simulations denoted as $S^i_{J,k}, i = I, \ldots, N_{J,k}$, we know from Lemma 2 that for every $S^i_{J,k}$, the error per macro step can be estimated via

$$\left\| \frac{\mathcal{E}^i_{J,k}(H_{J,k})}{H_{J,k}} \right\| = \mathcal{O}\left({H_{J,k}}^{\min\{p_{i_{J,k}}, q_{i_{J,k}}+1\}} \right) \tag{32}$$

with $p_{i_{J,k}}$ denoting the consistency order of the integration method of $S^i_{J,k}$ and $q_{i_{J,k}}$ the respective extrapolation order for external input values. With Corollary 1 follows for the consistency order of $S_{J,k}$

$$\left\| \frac{\mathcal{E}_{J,k}(H_{J,k})}{H_{J,k}} \right\| = \mathcal{O}\left(H_{J,k}^{\min\limits_{i=I,\dots,N_{J,k}} \{p_{i_{J,k}}, q_{i_{J,k}}+1\}} \right). \tag{33}$$

For every leaf simulation $S_{J,k}$ on level J with micro step size $h_{J,k}$, we obtain an estimate for the error per micro step with Lemma 1:

$$\left\| \frac{\mathcal{E}_{J,k}(h_{J,k})}{h_{J,k}} \right\| = \mathcal{O}\left(h_{J,k}^{\min\{p_{J,k}, q_{J,k}+1\}} \right) \tag{34}$$

As the indexing is unique, we can without confusion with some co-simulation declare $H_{J,k} := h_{J,k}$ and therefore in summary write the estimate for every simulation – cooperative as well as leaf simulation – on level J as

$$\left\| \frac{\mathcal{E}_{J,k}(H_{J,k})}{H_{J,k}} \right\| = \mathcal{O}\left(H_{J,k}^{\min\{p_{J,k}, q_{J,k}+1\}} \right) \tag{35}$$

when for co-simulation nodes, we define $p_{J,k} := \min\limits_{i=I,\dots,N_{J,k}} \{p_{i_{J,k}}\}$ and $q_{J,k} := \min\limits_{i=I,\dots,N_{J,k}} \{q_{i_{J,k}}\}$.

In the next step, we will assume this estimate for every simulation on a level $j+1$, $j \in \{1, \dots, J\}$:

$$\left\| \frac{\mathcal{E}_{j+1,k}(H_{j+1,k})}{H_{j+1,k}} \right\| = \mathcal{O}\left(H_{j+1,k}^{\min\{p_{j+1,k}, q_{j+1,k}+1\}} \right) \tag{36}$$

again with $H_{j+1,k} := h_{j+1,k}$ if $S_{j+1,k}$ is a leaf node and for co-simulation nodes $S_{j+1,k}$ defining $p_{j+1,k} := \min\limits_{i=I,\dots,N_{j+1,k}} \{p_{i_{j+1,k}}\}$ and $q_{j+1,k} := \min\limits_{i=I,\dots,N_{j+1,k}} \{q_{i_{j+1,k}}\}$ (using these definitions recursively in case for an i, the associated simulation $S_{j+1,k}^{i}$ $(= S_{j+2,l}$ for $l = i + \sum\limits_{m=1}^{k-1} N_{j+1,m})$ is again a co-simulation). Based on that, we consider the simulations on level j. For every leaf node on level j, Lemma 1 can directly be applied:

$$\left\| \frac{\mathcal{E}_{j,k}(h_{j,k})}{h_{j,k}} \right\| = \mathcal{O}\left(h_{j,k}^{\min\{p_{j,k}, q_{j,k}+1\}} \right), \tag{37}$$

which with $H_{j,k} := h_{j,k}$ can be written

$$\left\| \frac{\mathcal{E}_{j,k}(H_{j,k})}{H_{i_{j,k}}} \right\| = \mathcal{O}\left(H_{j,k}^{\min\{p_{j,k}, q_{j,k}+1\}} \right). \tag{38}$$

For every co-simulation on level j, we can utilize (36) and Corollary 1 to obtain

$$\left\| \frac{\mathcal{E}_{j,k}(H_{j,k})}{H_{j,k}} \right\| = \mathcal{O}\left(H_{j,k}^{\min\limits_{i=I,\dots,N_{j,k}} \{p_{i_{j,k}}, q_{i_{j,k}}+1\}} \right) = \mathcal{O}\left(H_{j,k}^{\min\{p_{j,k}, q_{j,k}+1\}} \right) \tag{39}$$

with $p_{j,k} := \min\limits_{i=I,\dots,N_{j,k}} \{p_{i_{j,k}}\}$ and $q_{j,k} := \min\limits_{i=I,\dots,N_{j,k}} \{q_{i_{j,k}}\}$ (recursively, if needed). Thus, with (38) we have

$$\left\| \frac{\mathcal{E}_{j,k}(H_{j,k})}{H_{j,k}} \right\| = \mathcal{O}\left(H_{j,k}^{\min\{p_{j,k}, q_{j,k}+1\}} \right) \tag{40}$$

for every cooperative and leaf simulation on level j.

This also holds for the topmost level $j = 1$, where only one co-simulation (and, naturally, no leaf node) remains. With $H := H_{1,1}$ and utilizing the fact that in this co-simulation, all $N = \sum_{j=1}^{J} \sum_{k=1}^{K_j} N_{j,k}$ participating leaf simulations and therefore, the consistency and extrapolation orders of every solution algorithm are finally considered, we obtain (30). □

3 Error Studies on a Coupled Three-Mass Oscillator

In the following, we consider an oscillator with three masses divided by force-displacement decomposition (see [13] for information on the coupling concept), which is illustrated in Fig. 5. With initial values and parameters given according to Table 1, we observe an increase of stiffnesses from left to right, which invites the introduction of another level of hierarchy.

Table 1. Initial values and parameter settings for the benchmark simulation.

x_1	x_2	x_3	$v_1 = v_2 = v_3$	c_1	c_{12}	c_{23}	c_3	d_1	d_{12}	d_{23}	d_3	m_1	m_2	m_3
1 m	2 m	3 m	0 m/s	1E−03 N/m	1E−01 N/m	10 N/m	100 N/m	0.1	0.4	1	2	10 kg	10 kg	10 kg

In a traditional co-simulation, Systems S^I, S^{II} and S^{III} would, in general, all be orchestrated by one algorithm demanding synchronization at the same time step. In a hierarchical approach, Systems S^{II} and S^{III} can be combined in a separate, lower-level co-simulation representing the new system $\widehat{S}^{II}$ that is co-simulated with System S^I on the top-level co-simulation.

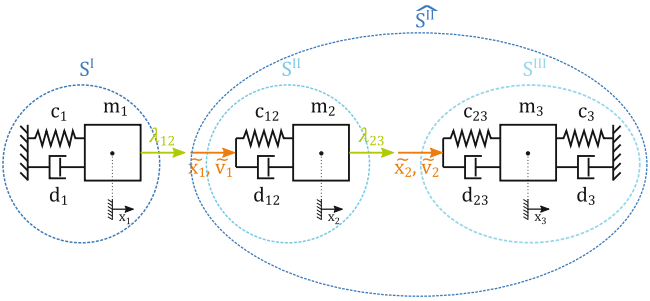

Fig. 5. Illustration of the hierarchical coupling of a three-mass oscillator.

The underlying equations can be interpreted as coupled Dahlquist equations, which invites investigations on stability by this example. Case studies that demonstrate the benefits of a hierarchical versus a traditional co-simulation approach regarding numerical stability are found in [9]. These include detailed

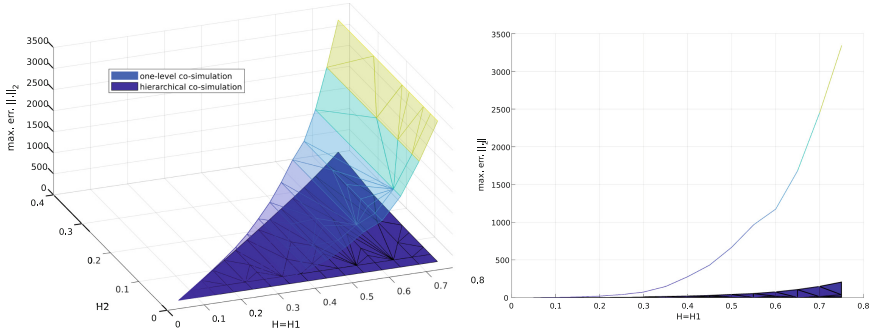

Fig. 6. Error ($\|.\|_2$ of all component errors) for the hierarchical and traditional co-simulation of the test scenario from $t_{start} = 0\,\text{s}$ to $t_{end} = 25\,\text{s}$ depending on macro step sizes.

tables comprising CPU time and errors for the simulation of the test scenario for 100 s, where we see that even for an upper-level macro step size twice as large ($H_1 = 0.2\,\text{s}$) as the one for the traditional co-simulation ($H = 0.1\,\text{s}$), the error can be reduced to less than one seventh if the second-level macro step size is chosen small enough ($H_2 = 0.05\,\text{s}$) while the elapsed computation time is barely increased (from 2.29 s to 2.58 s). Here, on the other hand, we will focus on the impact of varying macro step sizes on both co-simulation levels.

Since the differing stiffnesses result in slower and faster varying subsystems, the step sizes for the individual subsystem solvers are chosen accordingly with $h_I = 0.005\,\text{s}$, $h_{II} = 0.0025\,\text{s}$ and $h_{III} = 0.00125\,\text{s}$. Figure 6 shows the overall error – calculated by $\|.\|_2$ of the maximum errors of all states – depending on the macro step sizes $H = H_1$ for the traditional and upper-level co-simulation in the hierarchical approach, and H_2 for the second-level co-simulation in the hierarchical approach. The duration of all simulations is chosen with 25 s. H_2 ranges from 0.025 s over all multiples that are divisors of H_1 up to $H_1/2$ (for $H_2 = H_1$, the same results as for the traditional approach would be expected).

On the one hand, we immediately observe a faster ascent and more curvature for the error in the traditional approach. In addition, the impact of the choice of H_2 is clearly visible and comes out even more clearly in the separate illustration of the hierarchical approach in Fig. 7.

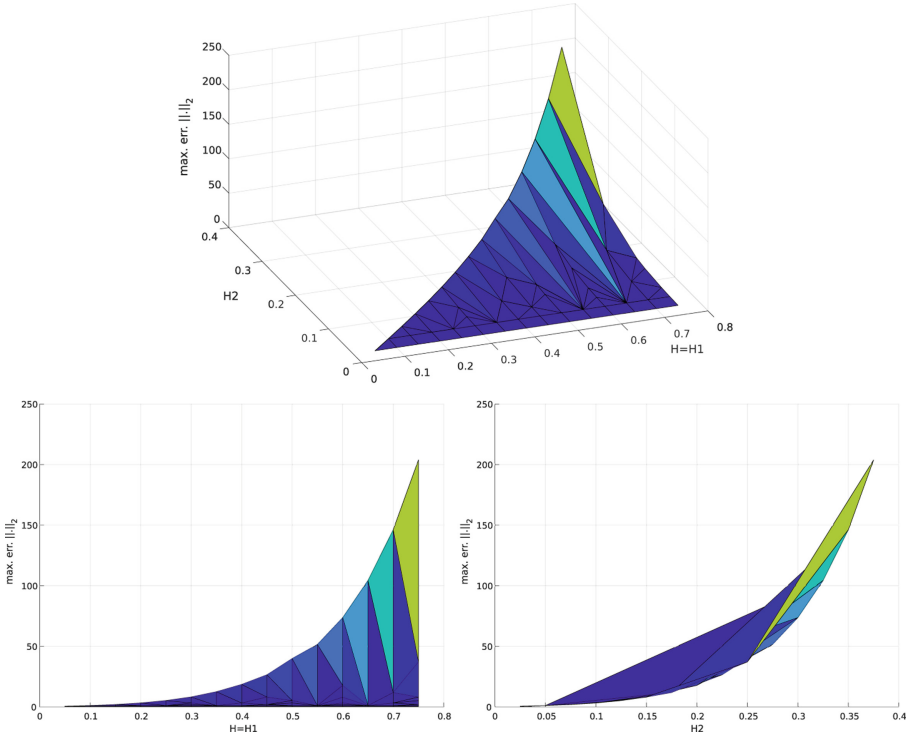

Fig. 7. Error ($\|.\|_2$ of all component errors) for the hierarchical co-simulation of the test scenario from $t_{start} = 0\,\mathrm{s}$ to $t_{end} = 25\,\mathrm{s}$ depending on macro step sizes.

4 Conclusion and Outlook

Above investigations show that consistency is maintained in hierarchical co-simulation, although it may potentially converge with lower order in comparison to the corresponding mono-simulation, depending on the extrapolation of external inputs. Since this is also the case for traditional co-simulation, no further loss of the order of consistency is added by the introduction of further hierarchies. On the contrary, as studies with varying macro step sizes show, error propagation is slowed down and accuracy increased if subsystems with closer dependencies are allowed to communicate more frequently while synchronization intervals with other subsystems can be increased.

Since hierarchical co-simulation is already permitted in certain frameworks and standards for co-simulation, the presented estimates along with investigations on stability in [8,9] provide the assertion that the application of hierarchical methods maintain and may even improve convergence.

Nevertheless, the method offers several aspects for further enhancement. Instead of parallel, non-iterative coupling algorithms with fixed macro step size, zero-order extrapolation and Euler integration methods used in the bench-

mark example from Sect. 3, strategies that are known to improve stability, performance, or accuracy for traditional co-simulation may be utilized in hierarchical co-simulation as well. Among these, the utilization of sequential, iterative or adaptive orchestration algorithms, different extrapolation orders and higher order and/or multistep subsystem solvers remain a topic for future investigations.

References

1. Arnold, M.: Multi-rate time integration for large scale multibody system models. In: Eberhard, P. (ed.) IUTAM Symposium on Multiscale Problems in Multibody System Contacts. No. 1 in IUTAM Bookseries, pp. 1–10. Springer, Netherlands (2007). https://doi.org/10.1007/978-1-4020-5981-0_1
2. Arnold, M.: Stability of sequential modular time integration methods for coupled multibody system models. J. Comput. Nonlinear Dyn. **5**(3), 031003 (2010). https://doi.org/10.1115/1.4001389
3. Arnold, M., Günther, M.: Preconditioned dynamic iteration for coupled differential-algebraic systems. BIT Numer. Math. **41**(1), 1–25 (2001). https://doi.org/10.1023/A:1021909032551
4. Arnold, M., Hante, S., Köbis, M.A.: Error analysis for co-simulation with force-displacement coupling. PAMM **14**(1), 43–44 (2014). https://doi.org/10.1002/pamm.201410014
5. Busch, M.: Zur effizienten Kopplung von Simulationsprogrammen. Kassel University Press (2012). https://books.google.at/books?id=0qBpXp-f2gQC
6. Esposito, J., Kumar, V.: Efficient dynamic simulation of robotic systems with hierarchy. In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), Seoul, South Korea, vol. 3, pp. 2818–2823. IEEE (2001). https://doi.org/10.1109/ROBOT.2001.933049
7. Günther, M., Rentrop, P.: Partitioning and multirate strategies in latent electric circuits. In: Bank, R.E., Gajewski, H., Bulirsch, R., Merten, K. (eds.) Mathematical Modelling and Simulation of Electrical Circuits and Semiconductor Devices, pp. 33–60. ISNM International Series of Numerical Mathematics, Birkhäuser, Basel (1994). https://doi.org/10.1007/978-3-0348-8528-7_3
8. Hafner, I.: Cooperative and multirate simulation: analysis, classification and new hierarchical approaches. Ph.D. thesis, TU Wien, Vienna, Austria (2021)
9. Hafner, I., Popper, N.: Investigation on stability properties of hierarchical co-simulation. SNE Simul. Notes Europe **31**(1), 17–24 (2021). https://doi.org/10.11128/sne.31.tn.10553
10. Hairer, E., Nørsett, S., Wanner, G.: Solving Ordinary Differential Equations I: Nonstiff Problems. Springer Series in Computational Mathematics, vol. 8. Springer, Heidelberg (1993). https://doi.org/10.1007/978-3-540-78862-1
11. Knorr, S.: Multirate-Verfahren in der co-simulation gekoppelter dynamischer Systeme mit Anwendung in der Fahrzeugdynamik. Diploma Thesis, Universität Ulm, Ulm, Germany, October 2002. http://www-num.math.uni-wuppertal.de/fileadmin/mathe/www-num/theses/da_knorr.pdf. Accessed 1 Oct 2021
12. Maten, J.T., Verhoeven, A., Guennouni, A.E., Beelen, T.: Multirate hierarchical time integration for electronic circuits. PAMM **5**(1), 819–820 (2005). https://doi.org/10.1002/pamm.200510381

13. Schweizer, B., Lu, D.: Semi-implicit co-simulation approach for solver coupling. Arch. Appl. Mech. **84**(12), 1739–1769 (2014). https://doi.org/10.1007/s00419-014-0883-5
14. Striebel, M.: Hierarchical mixed multirating for distributed integration of DAE network equations in chip design. Ph.D. thesis, Bergische Universität Wuppertal, April 2006
15. Tahir-Kheli, R.: Ordinary Differential Equations. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76406-1
16. Thule, C., et al.: Towards reuse of synchronization algorithms in co-simulation frameworks. In: Camara, J., Steffen, M. (eds.) Software Engineering and Formal Methods, pp. 50–66. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57506-9_5
17. Wang, J., Ma, Z.D., Hulbert, G.M.: A gluing algorithm for distributed simulation of multibody systems. Nonlinear Dyn. **34**(1), 159–188 (2003). https://doi.org/10.1023/B:NODY.0000014558.70434.b0
18. Zeidler, E. (ed.): Springer-Taschenbuch der Mathematik. Springer Fachmedien Wiesbaden, Wiesbaden (2013). http://link.springer.com/10.1007/978-3-8348-2359-5. Accessed 16 Feb 2016
19. Zeigler, B.P.: Object-Oriented Simulation with Hierarchical, Modular Models: Intelligent Agents and Endomorphic Systems. Elsevier Science, Saint Louis (2014). http://qut.eblib.com.au/patron/FullRecord.aspx?p=1876958. Accessed 2 Dec 2019
20. Zhang, H., Liang, S., Song, S., Wang, H.: Truncation error calculation based on Richardson extrapolation for variable-step collaborative simulation. Sci. China Inf. Sci. **54**(6), 1238–1250 (2011). https://doi.org/10.1007/s11432-011-4274-z