Reduction of Computation Time by Parallelization and Incorporating Co-Simulation Techniques

Jan Kraft, Bernhard Schweizer

Institute of Applied Dynamics
Technical University of Darmstadt
Otto-Berndt-Straße 2, 64287 Darmstadt, Germany
E-mail: kraft@ad.tu-darmstadt.de – Web page: www.ad.tu-darmstadt.de

Key words: Co-Simulation, Applied-Force Coupling, Dynamical Systems, Parallel Computing.

Abstract. The co-simulation methods considered here are based on the idea of splitting an overall model into different subsystems; the subsystems are then simulated as a coupled problem. Such an approach can be used advantageously for analyzing complex problems, for example the simulation of systems including different physical disciplines so that different specialized solvers have to be coupled in time domain. Another possible application of co-simulation methods, which is discussed here, concerns the parallelization of a monodisciplinary model.

1 INTRODUCTION

Co-simulation or solver coupling methods are used in various fields of applications. Examples can be found in [1] and [2]. The basic idea of co-simulation is to decompose an overall system into coupled subsystems. The formulation of the coupling conditions between the two subsystems depends on the considered problem. In the case of mechanical systems, coupling may be achieved by cutting through joints or by cutting through elements representing a physical force (torque). This leads to a coupling by reaction forces/torques [3][4] (constraint coupling) or to a connection by applied forces/torques (applied-force coupling). Co-simulation methods may further be subdivided into force/force-, force/displacement- and displacement/displacement-coupling approaches [5]. In this contribution, a force/force-decomposition approach is used and the subsystems are connected by nonlinear spring/damper-elements.

The methods presented here are weak coupling approaches, which implies that each subsystem is solved independently from the other subsystems within a macro-time step. Information (i.e. coupling variables) is only exchanged between the subsystems at certain communication-time points. The unknown coupling variables are approximated (extrapolated/interpolated) in the subsystems within a macro-time step. The separate integration of the subsystems is the crucial point for parallelizing the computation.

In this manuscript, two different numerical methods for solving the coupled problem are examined: an explicit co-simulation technique and a semi-implicit integration scheme. The semi-implicit method is based on a predictor/corrector procedure, where the corrector step is carried out only once.

2 CO-SIMULATION METHODS

To investigate the performance of the explicit and the semi-implicit co-simulation approach with regard to computation time, we use a nonlinear, large-scaled test model, which is described in the following subsection.

2.1 Nonlinear test model

One requirement for the test model is that it is straightforward to scale with respect to the number of degrees of freedom and with regard to the number of subsystems. Therefore, a series of n_K masses connected by nonlinear spring/damper-elements is used as test model.

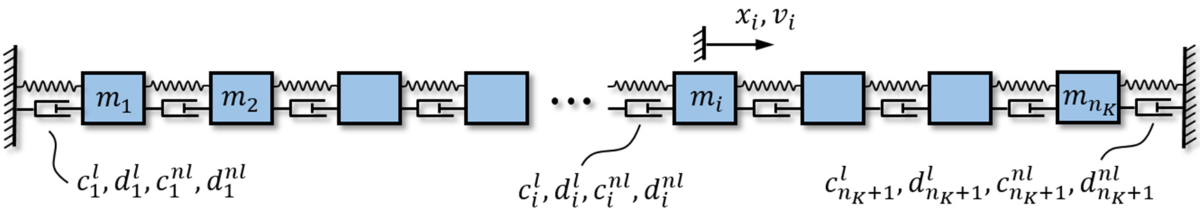
x_i, v_i
m_1
m_2
m_i
m_{n_K}
$c_1^l, d_1^l, c_1^{nl}, d_1^{nl}$
$c_i^l, d_i^l, c_i^{nl}, d_i^{nl}$
$c_{n_K+1}^l, d_{n_K+1}^l, c_{n_K+1}^{nl}, d_{n_K+1}^{nl}$

Figure 1: Nonlinear test model

Denoting the displacements of the oscillator-masses by the displacement coordinates x_i and the corresponding velocities by v_i, we obtain a system of $2n_K$ ordinary differential equations of the form

$$\begin{aligned} \dot{x}_i &= v_i \\ \dot{v}_i &= \frac{c_i^l}{m_i}(x_{i-1} - x_i) + \frac{d_i^l}{m_i}(\dot{x}_{i-1} - \dot{x}_i) + \frac{c_{i+1}^l}{m_i}(x_{i+1} - x_i) + \frac{d_{i+1}^l}{m_i}(\dot{x}_{i+1} - \dot{x}_i) \\ &\quad + \frac{c_i^{nl}}{m_i}(x_{i-1} - x_i)^3 + \frac{d_i^{nl}}{m_i}(\dot{x}_{i-1} - \dot{x}_i)^3 + \frac{c_{i+1}^{nl}}{m_i}(x_{i+1} - x_i)^3 + \frac{d_{i+1}^{nl}}{m_i}(\dot{x}_{i+1} - \dot{x}_i)^3 \end{aligned} \tag{1}$$

with $i = 1 \ldots n_K$. We assume that the chain is fixed at both ends ($x_0 = \dot{x}_0 = x_{n_K+1} = \dot{x}_{n_K+1} = 0$). The system parameters are the masses m_i, the linear stiffness coefficients c_i^l, the linear damping coefficients d_i^l, the nonlinear stiffness coefficients c_i^{nl} and the nonlinear damping coefficients d_i^{nl}. The values of the parameters used for the simulations are given in Table 1.

Table 1: Model parameters

c_i^l	$1.0e7$
d_i^l	1.0
c_i^{nl}	$1.0e9$
d_i^{nl}	$1.0e\text{-}4$
m_i	1.0

2.2 Decomposition of the test model

As mentioned above, the overall system is split into coupled subsystems by a force/force-decomposition approach [5]. This is achieved by cutting through certain nonlinear spring/damper-elements and by using the corresponding forces as coupling variables. The number of subsystems n is arbitrary, but usually it is much smaller than the number n_K of degrees of freedom of the overall system.

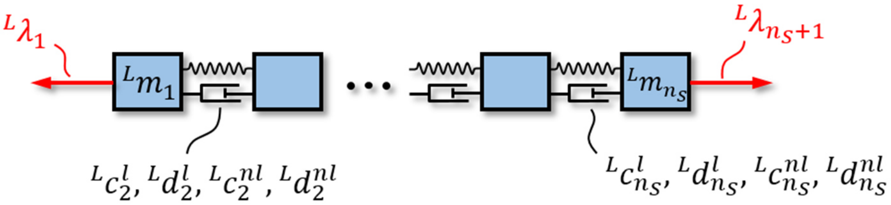
${}^L\lambda_1$
${}^L\lambda_{n_S+1}$
${}^Lm_1$
${}^Lm_{n_S}$
${}^Lc_2^l, {}^Ld_2^l, {}^Lc_2^{nl}, {}^Ld_2^{nl}$
${}^Lc_{n_S}^l, {}^Ld_{n_S}^l, {}^Lc_{n_S}^{nl}, {}^Ld_{n_S}^{nl}$

Figure 2: Arbitrary subsystem *L*

The set of n_S equations of motion for a subsystem reads as

$$\begin{aligned}
{}^L\dot{x}_j &= {}^Lv_j \\
{}^L\dot{v}_j &= \frac{{}^Lc_j^l}{{}^Lm_j}\left({}^Lx_{j-1} - {}^Lx_j\right) + \frac{{}^Ld_j^l}{{}^Lm_j}\left({}^L\dot{x}_{j-1} - {}^L\dot{x}_j\right) + \frac{{}^Lc_{j+1}^l}{{}^Lm_j}\left({}^Lx_{j+1} - {}^Lx_j\right) \\
&\quad + \frac{{}^Ld_{j+1}^l}{{}^Lm_j}\left({}^L\dot{x}_{j+1} - {}^L\dot{x}_j\right) + \frac{{}^Lc_j^{nl}}{{}^Lm_j}\left({}^Lx_{j-1} - {}^Lx_j\right)^3 + \frac{{}^Ld_j^{nl}}{{}^Lm_j}\left({}^L\dot{x}_{j-1} - {}^L\dot{x}_j\right)^3 \\
&\quad + \frac{{}^Lc_{j+1}^{nl}}{{}^Lm_j}\left({}^Lx_{j+1} - {}^Lx_j\right)^3 + \frac{{}^Ld_{j+1}^{nl}}{{}^Lm_j}\left({}^L\dot{x}_{j+1} - {}^L\dot{x}_j\right)^3 - \frac{{}^L\lambda_j}{{}^Lm_j} + \frac{{}^L\lambda_{j+1}}{{}^Lm_j}
\end{aligned} \tag{2}$$

with $j = 1 \ldots n_S$ and ${}^Lc_1^l = {}^Ld_1^l = {}^Lc_1^{nl} = {}^Ld_1^{nl} = {}^Lc_{n_S+1}^l = {}^Ld_{n_S+1}^l = {}^Lc_{n_S+1}^{nl} = {}^Ld_{n_S+1}^{nl} = 0$. The coupling forces are denoted by ${}^L\lambda_j$ and ${}^L\lambda_{j+1}$; they are only required for the coupling bodies ($j = 1$ and $j = n_S$) and are set to zero for the remaining bodies (${}^L\lambda_2 = \cdots = {}^L\lambda_{n_S} = 0$).

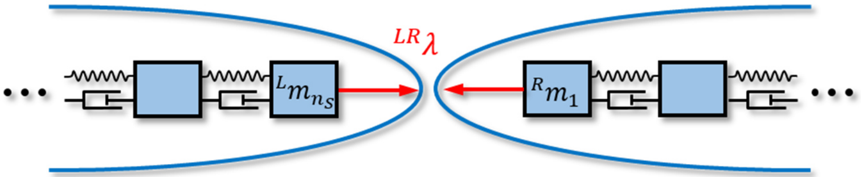

Figure 3: Coupling of two adjacent subsystems *L* and *R*

The coupling condition for two adjacent subsystems L and R reads (assuming that body n_S of subsystem L is coupled with body 1 of subsystem R)

$$\begin{aligned}
{}^{LR}g := {}^{LR}\lambda - {}^{LR}c_c^l\left({}^Rx_1 - {}^Lx_{n_S}\right) - {}^{LR}d_c^l\left({}^R\dot{x}_1 - {}^L\dot{x}_{n_S}\right) - {}^{LR}c_c^{nl}\left({}^Rx_1 - {}^Lx_{n_S}\right)^3 \\
- {}^{LR}d_c^{nl}\left({}^R\dot{x}_1 - {}^L\dot{x}_{n_S}\right)^3 = 0
\end{aligned} \tag{3}$$

with the coupling force ${}^{LR}\lambda = {}^L\lambda_{n_S+1} = {}^R\lambda_1$, the coupling parameters ${}^{LR}c_c^l$, ${}^{LR}d_c^l$, ${}^{LR}c_c^{nl}$ and ${}^{LR}d_c^{nl}$ and the state variables of the two coupling bodies.

2.3 Explicit co-simulation scheme

To solve the decomposed system as a coupled problem by using an explicit co-simulation method, the time interval is discretized with a macro-time grid. The macro-step size h_{mac} is chosen to be constant. Within an arbitrary macro-step from T_N to $T_{N+1} = T_N + h_{mac}$, each subsystem is integrated using extrapolation (interpolation) polynomials to approximate the coupling forces. After the subsystem integrations, the resulting states of the coupling bodies are substituted into the coupling condition to obtain the coupling force at the new macro-time point T_{N+1} (update of the coupling variables). The explicit co-simulation method has the advantage that a repetition of the macro-step is not necessarily required if a constant macro-step size is used. This may be a crucial point, if commercial subsystem solvers are used which often do not allow solver reintialization.

2.4 Semi-implicit co-simulation scheme

A detailed description of the implemented semi-implicit co-simulation procedure can be found in [6]. The basics of the approach are only briefly explained next.

The macro-time grid is assumed to be equidistant (macro-step size $h_{mac} = const.$). As mentioned above, the presented semi-implicit co-simulation method is based on a predictor/corrector approach with only one corrector step. An arbitrary co-simulation step from T_N to T_{N+1} is explained in the following subsections.

2.4.1 Predictor step

Within the predictor step, each subsystem is integrated twice from T_N to T_{N+1}: firstly with predicted (extrapolated) coupling variables (coupling forces) λ^p (${}^{LR}\lambda^p = {}^{L}\lambda^p_{ns+1} = {}^{R}\lambda^p_1$) and secondly with perturbed predicted coupling variables λ^Δ (${}^{LR}\lambda^\Delta = {}^{L}\lambda^\Delta_{ns+1} = {}^{R}\lambda^\Delta_1$). Note that for the reason of a concise representation, the subsystem indeces have been ommitted. The perturbed coupling variables are obtained by adding a small, user-defined perturbation to the predicted coupling variables, i.e. $\lambda^\Delta = \lambda^p + \Delta\lambda$.

With the predicted state variables $\boldsymbol{z}^p$ and the perturbed predicted states $\boldsymbol{z}^\Delta$ at T_{N+1}, the partial derivatives of the states with respect to the coupling variables can be approximated by finite differences

$$\left.\frac{\partial \boldsymbol{z}_c}{\partial \lambda}\right|_{\lambda^p} = \lim_{\Delta\lambda \to 0} \frac{\boldsymbol{z}_c(\lambda^p + \Delta\lambda) - \boldsymbol{z}_c(\lambda^p)}{\Delta\lambda} \approx \frac{\boldsymbol{z}_c^\Delta - \boldsymbol{z}_c^p}{\Delta\lambda}. \tag{4}$$

Note that partial derivatives only have to be calculated for the coupling bodies.

2.4.2 Corrector step

The approximated partial derivatives obtained in the predictor step are utilized to compute improved (corrected) coupling variables. Therefore, the coupling condition (3) is considered as a function of the coupling variable λ at T_{N+1} and expanded in a Taylor series. Choosing λ^p

as expansion point and neglecting higher-order terms $\mathcal{O}(\lambda^2)$, one obtains the linearized coupling condition

$$\begin{aligned}
g^{lin}(\lambda) &:= g(\lambda^p) + \left.\frac{\partial g}{\partial \lambda}\right|_{\lambda^p} (\lambda - \lambda^p) \\
&= \lambda^p - c_c^l\left({}^R x_1^p - {}^L x_{n_s}^p\right) - d_c^l\left({}^R v_1^p - {}^L v_{n_s}^p\right) - c_c^{nl}\left({}^R x_1^p - {}^L x_{n_s}^P\right)^3 \\
&\qquad - d_c^{nl}\left({}^R v_1^p - {}^L v_{n_s}^P\right)^3 \\
&\quad + \Bigg[1 - c_c^l\left(\left.\frac{\partial {}^R x_1^p}{\partial \lambda}\right|_{\lambda^p} - \left.\frac{\partial {}^L x_{n_s}^p}{\partial \lambda}\right|_{\lambda^p}\right) - d_c^l\left(\left.\frac{\partial {}^R v_1^p}{\partial \lambda}\right|_{\lambda^p} - \left.\frac{\partial {}^L v_{n_s}^p}{\partial \lambda}\right|_{\lambda^p}\right) \\
&\quad -3\, c_c^{nl}\left({}^R x_1^p - {}^L x_{n_s}^p\right)^2\left(\left.\frac{\partial {}^R x_1^p}{\partial \lambda}\right|_{\lambda^p} - \left.\frac{\partial {}^L x_{n_s}^p}{\partial \lambda}\right|_{\lambda^p}\right) \\
&\quad -3\, d_c^{nl}\left({}^R v_1^p - {}^L v_{n_s}^p\right)^2\left(\left.\frac{\partial {}^R v_1^p}{\partial \lambda}\right|_{\lambda^p} - \left.\frac{\partial {}^L v_{n_s}^p}{\partial \lambda}\right|_{\lambda^p}\right)\Bigg](\lambda - \lambda^p) = 0 \ .
\end{aligned} \tag{5}$$

Solving equation (5) for the coupling variable λ yields the corrected coupling force λ^c. In general, the predicted state variables and the predicted coupling force will not fulfill the coupling condition. The corrected coupling force (together with the corrected state variables), however, fulfills at least the linearized coupling condition (5). The subsystem integration within the corrector step is carried out by interpolating the coupling variables and making use of the corrected coupling forces λ_c.

The corrected state variables together with the corrected coupling force will in general not fulfill the nonlinear coupling conditions. To achieve consistent coupling forces, an update of the coupling forces at T_{N+1} is useful. Therefore, the corrected state variables of the coupling bodies are substituted into the coupling condition (3) in order to calculate updated coupling forces.

2.5 Subsystem solver

The subsystems are solved with the IDA solver from the SUNDIALS (Suite of Nonlinear and Differential/Algebraic Equation Solvers) package [7]. This implicit DAE solver is based on a variable-order variable-coefficient BDF implementation combined with either direct (sparse) or iterative methods for solving the linear system within the Newton iteration. For the present studies, the direct sparse linear solver is used.

2.6 Parallelized computation

Within a macro-step, each subsystem is integrated independently. Exchange of information takes only place before or after the integration processes. Therefore, all subsystems can be solved in parallel. The simulations have been carried out on a cluster so that all subsystem integrations could be fully parallelized.

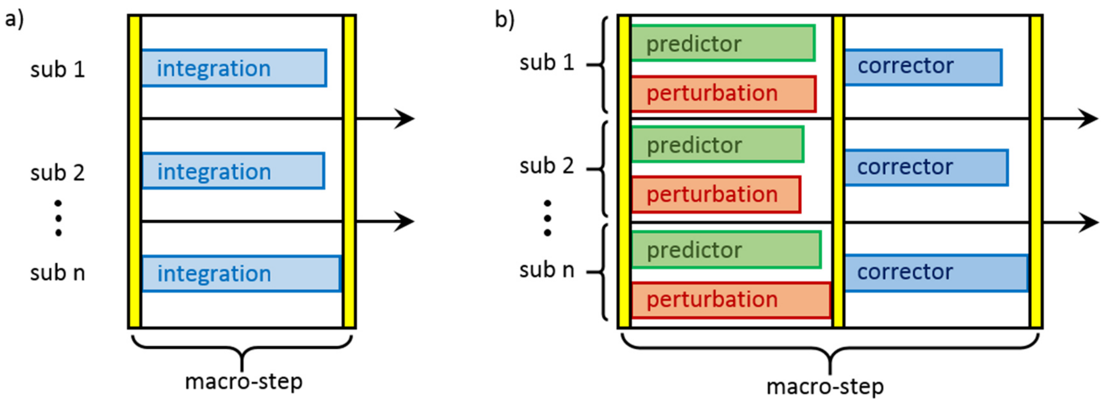

Figure 4: Parallelization scheme: a) explicit co-simulation and b) semi-implicit co-simulation

Applying a parallel implementation, the simulation time is usually strongly reduced. The computation time for the co-simulation can be estimated by

$$T_{cos}^{(expl)} \approx \frac{T_{mon}}{n^P} + C^{(expl)} \qquad \text{or} \qquad T_{cos}^{(semi)} \approx 2\frac{T_{mon}}{n^P} + C^{(semi)} \ , \tag{6}$$

where T_{mon} denotes the computation time of the monolithic model and n the number of subsystems. P represents the scaling factor of the computation time of the multibody implementation with respect to the number of degrees of freedom. For typical multibody systems, the value of P is between one and three, depending on the formulation of the equations of motion and the solving strategy. The overhead caused by the synchronization of parallel threads and additional calculations due to the co-simulation approach (e.g. solving equations (4) and (5) for the semi-implicit method) is summarized in the parameter C. The formula for T_{cos} implies the assumption that the overall system is split into similar subsystems so that the integration times for the different subsystems are similar.

3 SIMULATION RESULTS

In order to compare the co-simulation results with a reference solution, the normalized root mean square error (NRMSE) is computed for the state variables of the coupling bodies. The reference solution has been obtained by solving the monolithic model with a relative and absolute error tolerance of $1.0e$-12.

3.1 Convergence Analysis

The convergence behavior of the co-simulation methods is investigated by varying the (constant) macro-step size and by evaluating the local error of the coupling bodies. The test model used for the convergence analysis consists of $n_K = 144$ bodies and it is split into $n = 48$ equally sized subsystems. The model parameters are listed in Table 1. The analyses have been carried out for different approximation orders k of the coupling force. Also, two different error tolerances of the subsystem solver have been analyzed.

Fig. 5 shows the local error of the coupling bodies on position level. Two different subsystem error tolerances have been used, namely $1.0e$-12 (a) and $1.0e$-6 (b). The local

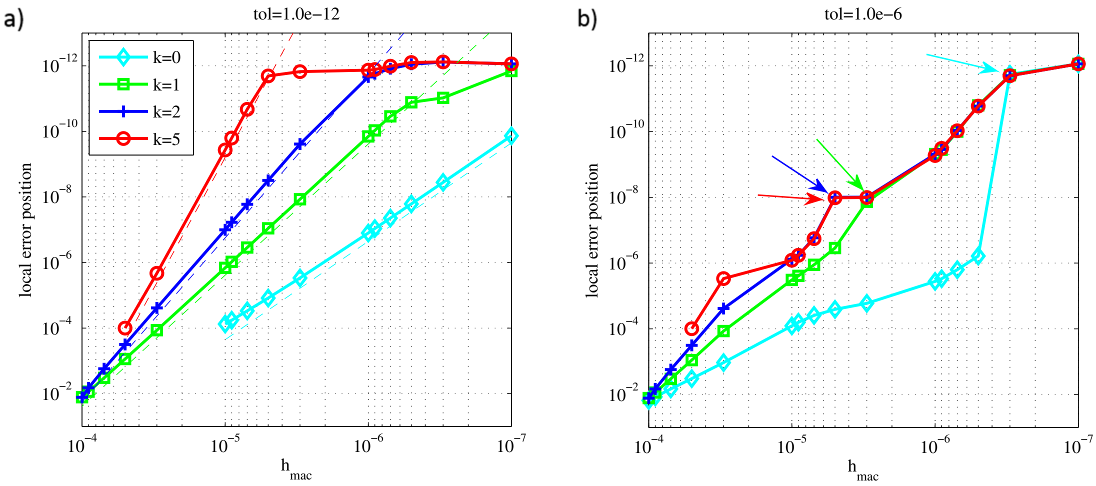

Figure 5: Convergence analysis for the semi-implicit co-simulation with different subsystem error tolerances: a) $1.0e$-12 and b) $1.0e$-6

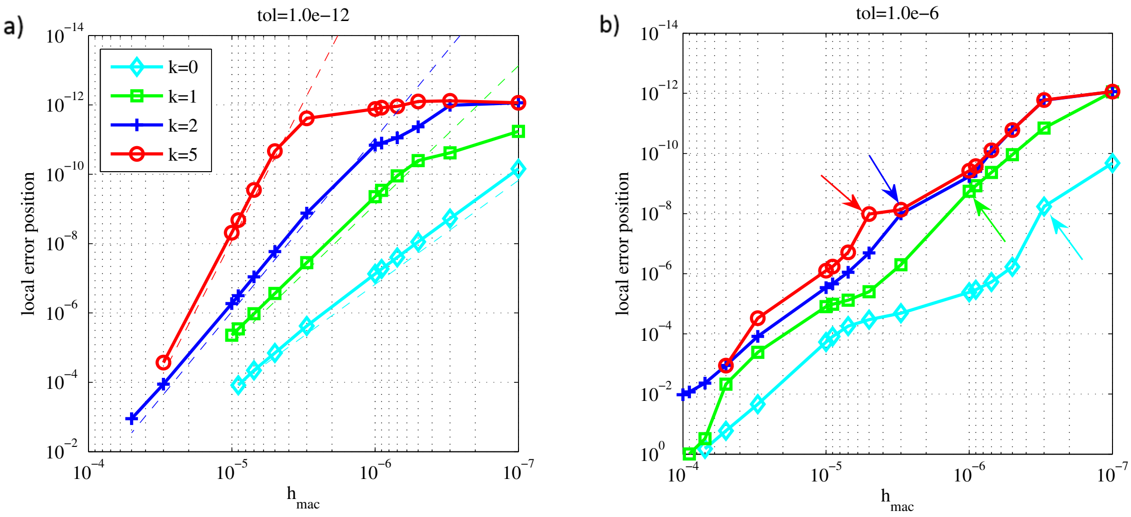

Figure 6: Convergence analysis for the explicit co-simulation with different subsystem error tolerances: a) $1.0e$-12 and b) $1.0e$-6

error of the co-simulation with the subsystem error tolerance $1.0e\text{-}12$ shows a convergence order of $\mathcal{O}(h_{mac}^{k+3})$. Errors with a magnitude of less than $1.0e\text{-}12$ are not resolved here because of the limited accuracy of the reference solution ($\text{tol}_{ref} = 1.0e\text{-}12$). The local error of the velocities (not shown here) converges with $\mathcal{O}(h_{mac}^{k+2})$. The results on the right hand side of Fig. 5 are influenced by inaccuracies of the subsystems solver. For small macro-step sizes, the order of interpolation becomes less relevant, because the subsystem solver takes usually only one integration step within each macro-step. The arrows indicate the point at which the macro-steps are of the same magnitude as the micro-steps (subsystem solver steps). Since the micro-step size is limited by the macro-step size here, the subsystem accuracy increases for small macro-step sizes.

The explicit co-simulation method shows a similar convergence behavior (Fig. 6) as the semi-implicit co-simulation approach.

3.2 Analysis of the computation time

The benefit of a parallel implementation with a co-simulation strategy can clearly be seen in the simulation of large-scaled models. Fig. 7 shows the computation times of models with a varying number of degrees of freedom; the plot has been generated with an explicit co-simulation method. The coupling forces were approximated by polynomials of order two. The co-simulation was carried out using a different number of subsystems. The black dashed line shows the resulting computation time of a monolithic simulation. The macro-step size was set to $5.0e\text{-}6$ so that the numerical error of the co-simulation is of the same magnitude as the error of the monolithic simulation. Because of the simple structure of the test model, the system matrix is very sparse. Therefore, the computation time scales almost linear with the number of degrees of freedom. The curves in Fig. 7 show that a co-simulation with seven subsystems reduces the computation time by over 33% compared to the monolithic simulation. A co-simulation with 127 subsystems turns out to be 16 times faster than the monolithic simulation. In a further study, not presented here in detail, a model with 2 million degrees of freedom was analyzed. Applying a semi-implicit co-simulation with 287 subsystems reduced the computation time by a factor of 120 compared to the monolithic simulation.

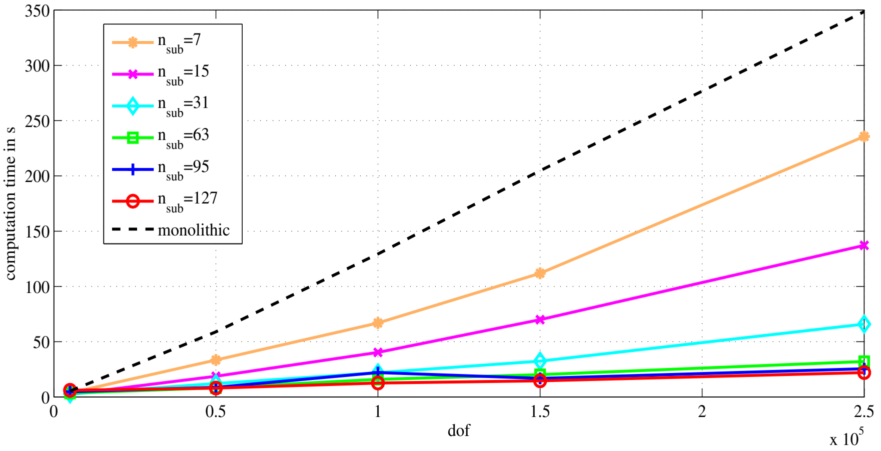

Figure 7: Computation time, explicit co-simulation

4 CONCLUSIONS

An explicit and a semi-implicit co-simulation approach have been used in order to parallelize a nonlinear dynamical model. The convergence behaviour of the two co-simulation methods has been investigated. It has been shown that the parallel implementation based on a solver coupling approach may significantly reduce the computation time compared to a monolithic model without increasing the numerical error markedly.

REFERENCES

[1] Ambrosio, J.; Pombo, J.; Pereira, M.; Antunes, P.; Mosca, A.: “A computational procedure for the dynamic analysis of the catenary-pantograph interaction in high-speed trains”, *Journal of Theoretical and Applied Mechanics*, 50/3, pp. 681-699, Warsaw, 2012.

[2] M. Naya, J. Cuadrado, D. Dopico, U. Lugris. An Efficient Unified Method for the Combined Simulation of Multibody and Hydraulic Dynamics: Comparison with Simplified and Co-Integration Approaches. *Archive of Mechanical Engineering*, Vol. LVIII, pp. 223–243, 2011.

[3] Kübler, R.; Schiehlen, W.: “Two methods of simulator coupling”, *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 6, pp. 93-113, 2000.

[4] Arnold, M.: “Stability of sequential modular time integration methods for coupled multibody system models”, *Journal of Computational and Nonlinear Dynamics*, Vol. 5, pp. 1-9, 2010.

[5] Schweizer, Bernhard, Pu Li, and Daixing Lu. "Explicit and Implicit Cosimulation Methods: Stability and Convergence Analysis for Different Solver Coupling Approaches." *Journal of Computational and Nonlinear Dynamics* 10.5 (2015): 051007.

[6] Schweizer, B.; Lu, D.: "Semi-implicit co-simulation approach for solver coupling." *Archive of Applied Mechanics* 84.12: 1739-1769, 2014.

[7] Hindmarsh, Alan C and Brown, Peter N and Grant, Keith E and Lee, Steven L and Serban, Radu and Shumaker, Dan E and Woodward, Carol S, and A. Collier: “SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers”, *ACM Transactions on Mathematical Software (TOMS)*, Vol. 31, No. 3, 2005.