Wiley
Online
Library
Early View publication on www.wileyonlinelibrary.com
(issue and page numbers not yet assigned;
citable using Digital Object Identifier – **DOI**)

ZAMM · Z. Angew. Math. Mech. 1–21 (2016) / **DOI** 10.1002/zamm.201500196

Continuous approximation techniques for co-simulation methods: Analysis of numerical stability and local error

Martin Busch*

Wiesenstraße 28, 91469 Hagenbüchach, Germany**

Received 24 July 2015, revised 11 November 2015, accepted 20 November 2015
Published online 25 January 2016

Key words Co-simulation, solver coupling, continuous approximation, extrapolated interpolation, stability, local error.

Coupling multiphysical systems by means of a co-simulation, the data between the subsystems are interchanged at a discrete macro time grid, also denoted as communication time grid. Between the communication points the coupling variables are approximated so that the numerical solvers in the subsystems can calculate the differential equations. Classical approximation techniques based on extrapolation entail a discontinuity in the equations in each macro time step which can slow down the numerical time integration. In the paper at hand a C^0-continuous technique for approximating the coupling variables is reconsidered and the numerical stability and the local error of the method are compared to the classical Lagrange approximation approach. Further, the method is enhanced to a C^1-continuous (continuous and differentiable) approximation technique. Both methods are investigated in combination with different numerical coupling approaches (sequential *Gauss-Seidel scheme*, parallel *Jacobi scheme*, *force/displacement coupling*, *displacement/displacement coupling*) which are commonly applied for co-simulation in technical applications. It is shown that the C^1-continuous approximation technique yields a similar numerical stability and a similar local error as the Lagrange approach which results in a comparable or even better overall performance (taking into account the advantage of continuity at the numerical calculation of the subsystem differential equations). Applying the C^0-continuous approach, a similar numerical stability is obtained. However, the order of the local error is significantly lower than for the C^1-continuous method and the Lagrange approach.

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

1 Introduction

In order to simulate coupled problems a co-simulation approach, which is also denoted as simulator coupling or coupled simulation, can be applied. Many techniques have been developed in this area in the past years. On the one hand these techniques can be distinguished regarding the decomposition of the overall system into subsystems, e.g.

- *Force-displacement coupling*: where one subsystem is kinematically excited and the other subsystem is force excited [1, 6, 8, 14, 15, 29]
- *Displacement-displacement coupling*: both subsystems are kinematically excited. The coupling force has to be calculated in each subsystem [8, 29].
- *Force-force coupling*: Both subsystems are excited by forces or accelerations[1]. Some well known *constraint coupling* approaches may be covered by this technique, see e.g. Refs. [5, 16, 20, 24, 31]. In order to calculate the algebraic constraint equations on acceleration level, either the Lagrange multipliers – which can be interpreted as reaction forces – or the accelerations have to be transferred between the subsystems, see [24].

On the other hand the co-simulation methods differ in the sequence of time integration and data interchange, e.g.

- *Jacobi scheme*: both subsystems are solved in parallel [16, 22, 26, 32].
- *Gauss-Seidel scheme*: the subsystems are solved in sequential order [13, 18, 22, 26, 27, 32].
- Fully iterative schemes, such as the *dynamic iteration* also known as *waveform relaxation* [25]: for the application of these methods, the subsystem solvers have to be reinitialized in the macro time steps [5, 24, 34].

* Corresponding author E-mail: busch@research-multibodydynamics.de

** The error and stability investigations for the Lagrange approximation approach, which is used as a comparative method in this paper, have been carried out at the former division *Multibody Systems* of the *Department of Mechanical Engineering, University of Kassel, D-34019 Kassel, Germany.*

[1] Accelerations may be regarded as forces according to Newton's second law.

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

- Semi-implicit coupling techniques, such as techniques applying a one-step iteration in the macro time steps [20,33] or techniques using the Jacobian matrix of the subsystems for stabilizing the coupled time integration [2,3,11,30].

A third differentiation can be made regarding the numerical approximation of the coupling variables during the macro steps. Many approaches use classical extrapolation/interpolation techniques, e.g. Taylor series expansion [22], Lagrange polynomials [9,19,21,29] or Hermite-polynomials [17,26]. A drawback of these classical approximation methods is that the coupling variables are discontinuous at the end of each macro step due to the update in the data interchange. Such discontinuities can slow down the time integration in the subsystems because the solvers have to reject their variable time steps more often. In this paper an approximation method denoted as *extrapolated interpolation* is reconsidered which is feasible to circumvent this discontinuity issue, see Sect. 2 for a detailed explanation. The idea of this method goes back to Dr. J. Rauh (Daimler AG) who was also involved in several works analyzing the potential of the method [6,14,23,28]. Even though the approach has been investigated in particular application examples, a fundamental analysis could not be found in the literature so far.

To obtain a functioning co-simulation, a numerical coupling approach has to satisfy two requirements. Firstly, the approach has to be consistent so that the local error vanishes with decreasing macro step size. Secondly, the approach has to be zero-stable, i.e. numerical errors are not gained through the macro steps if the macro step size H vanishes to zero [5,24]. While zero-stability is necessary for convergence, it is not sufficient to describe the efficiency of a co-simulation approach. Efficiency is related with the numerical stability, i.e. the error gain for finite macro step sizes $H > 0$. The more numerical stable a co-simulation method is, the larger macro steps can be used and the faster the co-simulation can be carried out. As in the case of standard time integration methods, the investigation of numerical stability requires a test model. While for standard time integration methods Dahlquists test equation [12] - describing a one mass oscillator - is often used, for coupled problems a two-mass oscillator is advantageous [8,29,30]. For the above mentioned co-simulation methods *Gauss-Seidel scheme*, *Jacobi scheme* and *dynamic iteration* using Lagrange polynomial approximation the numerical stability has been investigated for *force-displacement* and *displacement-displacement couplings* in Ref. [7,8]. In Ref. [7,10] the coupling approaches are considered for a Hermite polynomial approximation. The numerical stability of semi-implicit coupling methods in combination with *constraint couplings* can for instance be found in Ref. [31].

In the current paper the numerical stability and the local error of the extrapolated interpolation method is investigated using the two-mass oscillator. The calculations are carried out for co-simulation approaches of *Jacobi* and *Gauss-Seidel* type in combination with the widely used decomposition techniques *force-displacement coupling* and *displacement-displacement coupling*. Only the coupling via applied forces is considered, couplings based on algebraic constraints are not investigated here. The results are compared to the classical polynomial approximation. The continuous approximation methods are explained in Sect. 2. In Sect. 3 the test model used for the investigation is introduced. Further, the equations required for the stability analysis are derived. The stability results and the local error plots are discussed in Sect. 4.

2 C^0- and C^1-continuous approximation method: *extrapolated interpolation*

The present paper follows the block oriented framework of co-simulation according to Ref. [24] where the overall system can be represented by the subsystem equations

$$\underline{\dot{z}}_1 = \underline{f}_1(\underline{z}_1, \underline{u}_1) \qquad \underline{\dot{z}}_2 = \underline{f}_2(\underline{z}_2, \underline{u}_2) \tag{1a}$$

$$\underline{y}_1 = \underline{g}_1(\underline{z}_1, \underline{u}_1) \qquad \underline{y}_2 = \underline{g}_2(\underline{z}_2, \underline{u}_2) \tag{1b}$$

and the coupling conditions

$$\begin{pmatrix} \underline{u}_1 \\ \underline{u}_2 \end{pmatrix} = \begin{pmatrix} \underline{\underline{0}} & \underline{\underline{I}} \\ \underline{\underline{I}} & \underline{\underline{0}} \end{pmatrix} \cdot \begin{pmatrix} \underline{y}_1 \\ \underline{y}_2 \end{pmatrix} \tag{1c}$$

which connect the output of one subsystem with the input of the other subsystem. In Eq. (1) the vector-valued functions $\underline{f}_1$, $\underline{f}_2$ denote the right hand sides of the internal differential equations and $\underline{g}_1$, $\underline{g}_2$ the output functions of the subsystems. Further, the internal state vectors are denoted by $\underline{z}_1$, $\underline{z}_2$ and the subsystem inputs and the outputs are represented by the vectors $\underline{u}_1$, $\underline{u}_2$ and $\underline{y}_1$, $\underline{y}_2$. To keep the notation simple, the presentation is restricted to $r = 2$ coupled subsystems but the here-considered approximation techniques as well as the basic results on stability and convergence remain valid for $r > 2$ as well.

Applying a co-simulation, the subsystems can be solved independently which is also denoted as weak coupling approach. Using e.g. a *Gauss-Seidel scheme* for generating a numerical solution, the subsystems are solved in sequential order and the coupling variables have to be approximated. As mentioned in the introduction, using classical polynomial extrapolation techniques for the approximation can be disadvantageous, because the coupling variables undergo a discontinuity after each

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

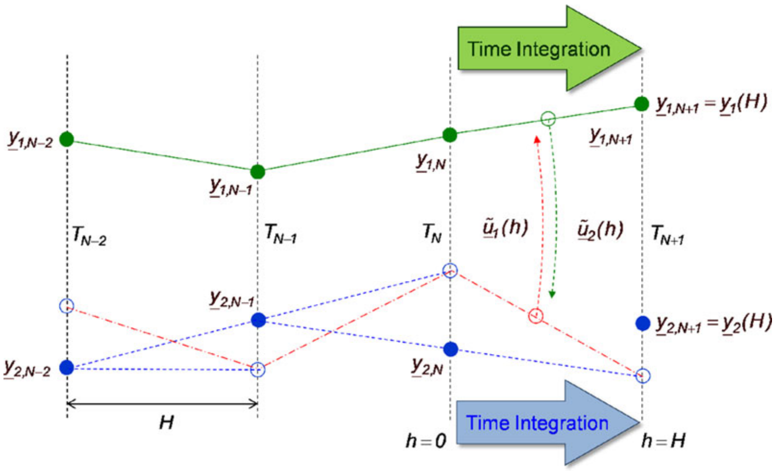


Fig. 1 Co-simulation carried out with a *Gauss-Seidel scheme* at macro time step $T_N \rightarrow T_{N+1}$ using step size H: An extrapolated interpolation method (red dash-dotted curve) is used for obtaining continuous input vectors $\underline{\tilde{u}}_1(h)$ in the micro time steps of subsystem 1, based on output vectors $\underline{y}_2$ of subsystem 2. Alternatively, a classical polynomial extrapolation (blue dashed curve) would entail a discontinuity at each macro time point T_i in the inputs. The input vectors $\underline{\tilde{u}}_2(h)$ for the micro time steps in subsystem 2 are continuously interpolated from the output vectors $\underline{y}_1$ of subsystem 1 (green solid curve) as it is common for a sequential *Gauss-Seidel scheme*.

macro time step. This behavior can be seen in Fig. 1 in the blue dashed curve. Carrying out a macro time step from T_N to T_{N+1} with the step size H, the solver in subsystem 1 calculates the output vector $\underline{y}_{1,N+1}$. Thereto it requires coupling variables as subsystem input $\underline{u}_1$ which are generated from the subsystem outputs $\underline{y}_2$ of subsystem 2. Since the data interchange between the subsystems is only accomplished at the discrete macro time points T_i, the inputs have to be approximated during the time integration. The solution $\underline{y}_{2,N+1}$ at T_{N+1} is not yet computed in subsystem 2, hence, an extrapolation from the previous solutions $\underline{y}_{2,N}$, $\underline{y}_{2,N-1}$, ... is used to calculate an approximated input vector $\underline{\tilde{u}}_1$. Using a classical polynomial extrapolation (blue dashed curve), the approximated coupling variables in the intervals $[T_{N-1}, T_N]$ and $[T_N, T_{N+1}]$ are not continuous so that the subsystem solver observes a sudden change in the right hand side at each T_i. As a result, the time-integration step size will be reduces at T_i in order to meet the solver tolerance. Since the discontinuity cannot be resolved by step size reduction within the subsystems, large numerical effort or even an abort of the time integration may occur.

One approach to avoid this behavior is the so-called *extrapolated interpolation* method [6, 14, 23, 28], see the red dash-dotted curve in Fig. 1. In each macro time window a classical polynomial extrapolation (blue dashed curve) is carried out in order to obtain discrete coupling values at T_i (blue circles). In the current macro time window $[T_N, T_{N+1}]$ these extrapolated values are smoothed by means of a polynomial interpolation (red dash-dotted curve) which is continuous at the end of the macro time window. In the C^0-version of this approach, which means that the approximation method is continuous but not differentiable, the interpolation polynomial is linear.

However, C^0-continuity of the input variables is not sufficient for a converging numerical time integration in the subsystems. Since implicit solvers require the Jacobian matrix of the right hand side, the derivative of the right hand side with respect to the subsystem inputs is taken into account. Hence, at least C^1-continuity is needed for the coupling variables so that the Jacobian matrix exists across the end of the current macro time window. For demanding C^1-continuity of the coupling variables the approach in Fig. 1 can simply be enhanced, see Fig. 2. Thereto, the smoothing in the current macro time window is accomplished by means of a cubic Hermite polynomial interpolation. The Hermite polynomial depends on the extrapolated values at T_i (blue circles) and also on the time derivative of the underlying extrapolation polynomials at T_i (blue arrows). It should be remarked that the C^0-version and the C^1-version of the extrapolated interpolation method depend on the same sampling points, however, the approximated input vectors differ.

Using a sequential *Gauss-Seidel scheme* for the co-simulation, the coupling variables for the second subsystem $\underline{\tilde{u}}_2$ can be calculated using a classical polynomial interpolation, see the green solid curve in Figs. 1 and 2, since the output vector $\underline{y}_{1,N+1}$ of subsystem 1 is already available at T_{N+1} in the current macro time window. With the approximated coupling variables the output vector $\underline{y}_{2,N+1}$ at T_{N+1} can be calculated in subsystem 2 and the macro step is finished. Applying a parallel *Jacobi scheme* instead of a sequential *Gauss-Seidel scheme*, the mentioned extrapolation-based approaches have to be applied in both subsystems.

For the sake of completeness, in Appendix A – see also Wiley online library – the approximation schemes for the C^0- and C^1-continuous extrapolated interpolation method are shown using constant and quadratic polynomials for the underlying extrapolation. The variables in the figures are denoted in the same manner as in Figs. 1 and 2.

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

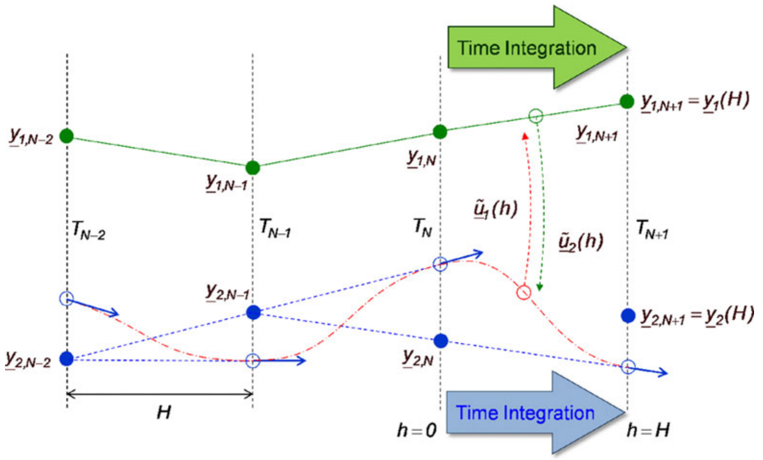


Fig. 2 Similar approach as in Fig. 1. However, a cubic Hermite spline is used for smoothing the coupling variable $\underline{\tilde{u}}_1(h)$ in the current macro time step $T_N \to T_{N+1}$. The cubic Hermite polynomial is based on the extrapolated values at T_N and T_{N+1} and the corresponding time derivatives of the polynomials (blue arrows).

3 Detailed investigation of the continuous approximation methods

3.1 Test model for stability analysis

As mentioned in the introduction, a 2-dof test model is used to investigate the numerical stability of the co-simulation methods. This test model can be regarded as a 2-mass oscillator, see Fig. 3, with the position coordinates x_1 and x_2 of the two masses. The constant parameters m_1, m_2, c_1, c_2 and d_1, d_2 denote mass, stiffness and damping coefficients. The parameters c_k and d_k denote the coupling stiffness and damping coefficient for the coupling spring between the two masses. The equation of motion of the 2-mass oscillator is given in state-space vector form

$$\dot{\underline{z}} = \underline{\underline{A}} \cdot \underline{z}, \quad \underline{z} = \begin{pmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{pmatrix} \in \mathbb{R}^4 \text{ and } \underline{\underline{A}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\dfrac{c_1 + c_k}{m_1} & -\dfrac{d_1 + d_k}{m_1} & \dfrac{c_k}{m_1} & \dfrac{d_k}{m_1} \\ 0 & 0 & 0 & 1 \\ \dfrac{c_k}{m_2} & \dfrac{d_k}{m_2} & -\dfrac{c_k + c_2}{m_2} & -\dfrac{d_k + d_2}{m_2} \end{pmatrix} \in \mathbb{R}^{4\times 4} \tag{2}$$

with the state matrix $\underline{\underline{A}}$. Applying some initial conditions $\underline{z}(t_0) = \underline{z}_0$, the solution of Eq. (2) is given by the vector function

$$\underline{z}(t) = \exp\left[\underline{\underline{A}}(t - t_0)\right] \cdot \underline{z}_0 \tag{3}$$

where $\exp[\underline{\underline{X}}] = \sum\limits_{k=0}^{\infty} \dfrac{1}{k!} (\underline{\underline{X}})^k$ denotes the matrix-exponential function.

3.2 Decomposition of the 2-DOF test model into subsystems

Splitting the overall test model into subsystems results in the following equations in state-space form

$$\dot{\underline{z}}_1 = \underline{\underline{A}}_1 \cdot \underline{z}_1 + \underline{\underline{B}}_1 \cdot \underline{u}_1 \qquad \dot{\underline{z}}_2 = \underline{\underline{A}}_2 \cdot \underline{z}_2 + \underline{\underline{B}}_2 \cdot \underline{u}_2 \tag{4a}$$

$$\underline{y}_1 = \underline{\underline{C}}_1 \cdot \underline{z}_1 + \underline{\underline{D}}_1 \cdot \underline{u}_1 \qquad \underline{y}_2 = \underline{\underline{C}}_2 \cdot \underline{z}_2 + \underline{\underline{D}}_2 \cdot \underline{u}_2 \tag{4b}$$

$$\begin{pmatrix} \underline{u}_1 \\ \underline{u}_2 \end{pmatrix} = \begin{pmatrix} \underline{\underline{0}} & \underline{\underline{I}} \\ \underline{\underline{I}} & \underline{\underline{0}} \end{pmatrix} \cdot \begin{pmatrix} \underline{y}_1 \\ \underline{y}_2 \end{pmatrix} \tag{4c}$$

with $\underline{z}_1 = \begin{pmatrix} x_1 \\ \dot{x}_1 \end{pmatrix} \in \mathbb{R}^2$, $\underline{z}_2 = \begin{pmatrix} x_2 \\ \dot{x}_2 \end{pmatrix} \in \mathbb{R}^2$ the internal states and $\underline{u}_1$, $\underline{u}_2$, $\underline{y}_1$, $\underline{y}_2 \in \mathbb{R}^2$ the input and output vectors of the subsystems. The precise form of the subsystem matrices $\underline{\underline{A}}_i$, $\underline{\underline{B}}_i$, $\underline{\underline{C}}_i$, and $\underline{\underline{D}}_i$ depends on the decomposition approach,

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

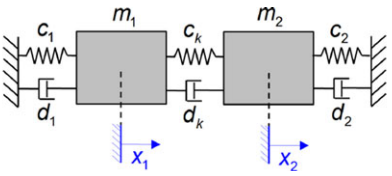
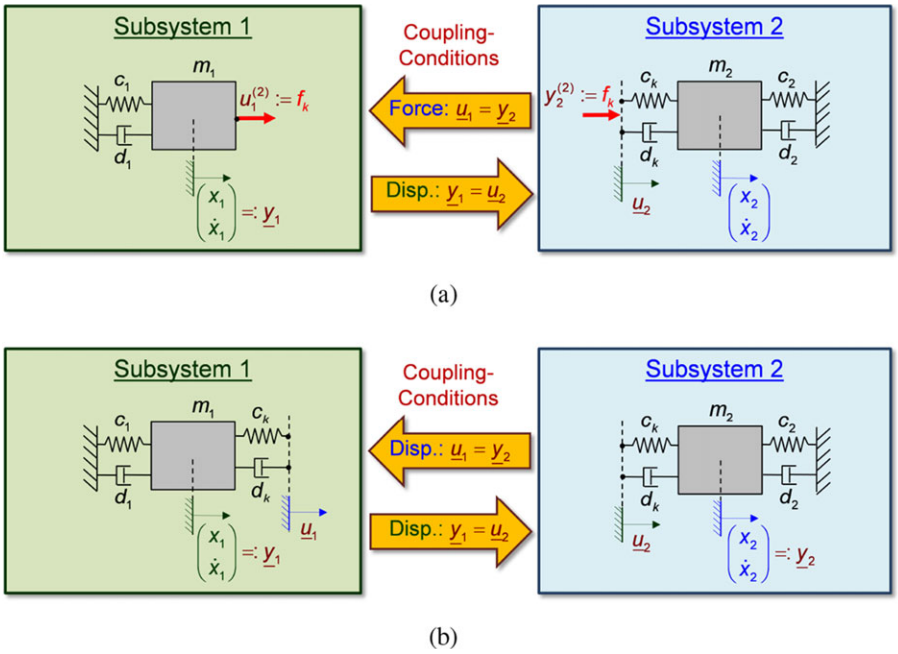


Fig. 3 Test model for co-simulation: a linear 2-dof oscillator.


Subsystem 1
Coupling-Conditions
Subsystem 2
m_1
c_1
d_1
$u_1^{(2)} := f_k$
Force: $\underline{u}_1 = \underline{y}_2$
Disp.: $\underline{y}_1 = \underline{u}_2$
$y_2^{(2)} := f_k$
c_k
m_2
c_2
d_k
d_2
$\begin{pmatrix} x_1 \\ \dot{x}_1 \end{pmatrix} =: \underline{y}_1$
$\underline{u}_2$
$\begin{pmatrix} x_2 \\ \dot{x}_2 \end{pmatrix}$
Subsystem 1
Coupling-Conditions
Subsystem 2
m_1
c_1
c_k
d_1
d_k
u_1
Disp.: $\underline{u}_1 = \underline{y}_2$
Disp.: $\underline{y}_1 = \underline{u}_2$
$\begin{pmatrix} x_1 \\ \dot{x}_1 \end{pmatrix} =: \underline{y}_1$
c_k
m_2
c_2
d_k
d_2
$\underline{u}_2$
$\begin{pmatrix} x_2 \\ \dot{x}_2 \end{pmatrix} =: \underline{y}_2$

(a)

(b)

Fig. 4 Modular modeling of 2-DOF oscillator: *Force/displacement coupling* (a). *Displacement/displacement coupling* (b).

see Fig. 4. The calculation of the matrices for the *force-displacement coupling* (one subsystem is force excited and one subsystem is kinematically excited) and *displacement-displacement coupling* (both subsystems are kinematically excited) can be found in Ref. [8,10], for instance. For the analyzes carried out in the current paper a compact form of the equations is sufficient. Note however that for both decomposition approaches at least one matrix $\underline{\underline{D}}_1$ or $\underline{\underline{D}}_2$ is a zero matrix which is necessary to ensure zero-stability of the co-simulation [7,24].

3.3 Calculation of numerical stability and local error of the co-simulation

For investigating the numerical stability of the co-simulation method, one macro time step $T_N \to T_{N+1}$ is regarded in detail. Since a co-simulation is a weak coupling approach, the analytical coupling conditions (4c) are relaxed and the subsystem input vectors are replaced by the approximated input vectors $\tilde{\underline{u}}_1^{(k)}(t)$, $\tilde{\underline{u}}_2^{(j)}(t)$ of degree k and j

$$\dot{\underline{z}}_1 = \underline{\underline{A}}_1 \cdot \underline{z}_1 + \underline{\underline{B}}_1 \cdot \tilde{\underline{u}}_1^{(k)}(t) \qquad \dot{\underline{z}}_2 = \underline{\underline{A}}_2 \cdot \underline{z}_2 + \underline{\underline{B}}_2 \cdot \tilde{\underline{u}}_2^{(j)}(t) \tag{5a}$$

$$\underline{y}_1 = \underline{\underline{C}}_1 \cdot \underline{z}_1 + \underline{\underline{D}}_1 \cdot \tilde{\underline{u}}_1^{(k)}(t) \qquad \underline{y}_2 = \underline{\underline{C}}_2 \cdot \underline{z}_2 + \underline{\underline{D}}_2 \cdot \tilde{\underline{u}}_2^{(j)}(t) \tag{5b}$$

For analyzing the macro-time step $T_N \to T_{N+1}$ the time variable $t \in [T_N, T_{N+1}]$ is transformed into a step size variable $h \in [0, H]$ by means of the substitution $h = t - T_N$. The variable $H := T_{N+1} - T_N$ then denotes the macro-step size. The state equations of both subsystems can be combined into one single inhomogeneous ODE system

$$\underbrace{\begin{pmatrix} \dot{\underline{z}}_1 \\ \dot{\underline{z}}_2 \end{pmatrix}(h)}_{\dot{\underline{z}}(h)} = \underbrace{\begin{pmatrix} \underline{\underline{A}}_1 & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{A}}_2 \end{pmatrix}}_{\tilde{\underline{\underline{A}}}} \cdot \underbrace{\begin{pmatrix} \underline{z}_1 \\ \underline{z}_2 \end{pmatrix}(h)}_{\underline{z}(h)} + \underbrace{\begin{pmatrix} \underline{\underline{B}}_1 & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{B}}_2 \end{pmatrix}}_{\tilde{\underline{\underline{B}}}} \cdot \underbrace{\begin{pmatrix} \tilde{\underline{u}}_1^{(k)} \\ \tilde{\underline{u}}_2^{(j)} \end{pmatrix}(h)}_{\tilde{\underline{u}}(h)} , \tag{6}$$

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

with an analytical solution given by

$$\underline{z}(h) = \exp\left[\underline{\underline{\tilde{A}}} \cdot h\right] \cdot \left(\int_0^h \exp\left[-\underline{\underline{\tilde{A}}} \cdot \tau\right] \cdot \underline{\underline{\tilde{B}}} \cdot \underline{\tilde{u}}(\tau) d\tau + \underline{z}_N \right) \quad \text{with} \quad \underline{z}_N := \underline{z}(h = 0). \tag{7}$$

Evaluating the solution at the macro-step size H, i.e. at the new macro time point T_{N+1}, one obtains the state vector $\underline{z}_{N+1} := \underline{z}(h = H)$. Evaluating the output vectors at T_{N+1} and combining it into a single equation gives

$$\underbrace{\begin{pmatrix} \underline{y}_{1,N+1} \\ \underline{y}_{2,N+1} \end{pmatrix}}_{\underline{y}_{N+1}} = \underbrace{\begin{pmatrix} \underline{\underline{C}}_1 & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{C}}_2 \end{pmatrix}}_{\underline{\underline{\tilde{C}}}} \cdot \underline{z}_{N+1} + \underbrace{\begin{pmatrix} \underline{\underline{D}}_1 & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{D}}_2 \end{pmatrix}}_{\underline{\underline{\tilde{D}}}} \cdot \underline{\tilde{u}}(H). \tag{8}$$

The different numerical coupling approaches can now be distinguished by specifying the input polynomials. It is advantageous to represent the polynomials in a Lagrange form. The Lagrange and corresponding Hermite basis polynomials of degree k and $2k + 1$ are defined as

$$L_m^{(k)}(h) := \begin{cases} 0 & \text{if } m < 0 \\ 0 & \text{if } m > k \\ \prod\limits_{\substack{n=0 \\ n \neq m}}^{k} \dfrac{h - h_n}{h_m - h_n} & \text{otherwise} \end{cases} \quad \text{and} \tag{9}$$

$$H_{m,0}^{(k)}(h) := \left[L_m^{(k)}(h)\right]^2 \cdot \left(1 - 2 \cdot \frac{d}{dh} L_m^{(k)}(h)\bigg|_{h = h_m} \cdot (h - h_m)\right),$$

$$H_{m,1}^{(k)}(h) := \left[L_m^{(k)}(h)\right]^2 \cdot (h - h_m)$$

with respect to the equidistant sampling points $h_n := -(k - n) \cdot H$ for $n = 0, \ldots, k$, i.e. the macro time points where the communication between the subsystems takes place. For the special case of constant polynomials ($k = 0$), the empty product in (9) yields $L_0^{(0)}(h) \equiv 1$. The Hermite basis polynomials satisfy the interpolation conditions

$$\frac{d^\nu}{dh^\nu} H_{m,\mu}^{(k)}(h)\bigg|_{h = h_n} = \begin{cases} 1 & \text{if } \nu = \mu \wedge n = m \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

Using a classical polynomial extrapolation $\underline{\Phi}_1$ for the inputs of subsystem 1, the output data $\underline{y}_{2,N-k}, \ldots, \underline{y}_{2,N}$ of subsystem 2 at the $k + 1$ previous macro-time points are applied according to

$$\begin{aligned} \underline{\tilde{u}}_1^{(k)}(h) &:= \underline{\Phi}_1\left(\left[-k \cdot H, \ldots, -H, 0\right], \left[\underline{y}_{2,N-k}, \ldots, \underline{y}_{2,N-1}, \underline{y}_{2,N}\right]; h\right) \\ &= \sum_{i=0}^{k} \underline{y}_{2,N-i} \cdot L_{k-i}^{(k)}(h) \qquad \text{(Classical polynomial extrapolation in subsystem 1).} \end{aligned} \tag{11}$$

Alternatively, an extrapolated interpolation method can be applied, see Fig. 1, using the outputs of subsystem 2 at the $k + 2$ previous macro-time points

$$\begin{aligned} \underline{\Phi}_{1,N} &:= \underline{\Phi}_1\left(\left[-(k+1) \cdot H, \ldots, -2H, -H\right], \left[\underline{y}_{2,N-(k+1)}, \ldots, \underline{y}_{2,N-2}, \underline{y}_{2,N-1}\right]; h = 0\right) \\ &= \sum_{i=0}^{k} \underline{y}_{2,N-1-i} \cdot L_{k-i}^{(k)}(h - -H)\bigg|_{h = 0} \qquad \text{(Classical polynomial extrapolation at } T_N\text{)} \\ \underline{\Phi}_{1,N+1} &:= \underline{\Phi}_1\left(\left[-k \cdot H, \ldots, -H, 0\right], \left[\underline{y}_{2,N-k}, \ldots, \underline{y}_{2,N-1}, \underline{y}_{2,N}\right]; h = H\right) \\ &= \sum_{i=0}^{k} \underline{y}_{2,N-i} \cdot L_{k-i}^{(k)}(h)\bigg|_{h = H} \qquad \text{(Classical polynomial extrapolation at } T_{N+1}\text{)} \\ \underline{\tilde{u}}_1^{(k)}(h) &:= \underline{\Phi}_{1,N} \cdot L_0^{(1)}(h - H) + \underline{\Phi}_{1,N+1} \cdot L_1^{(1)}(h - H) \quad (C^0\text{-approximation in subsystem 1).} \end{aligned} \tag{12a}$$

As can be seen, a classical polynomial extrapolation $\underline{\Phi}_1$ of degree k is used to obtain the two discrete data points $\underline{\Phi}_{1,N}$ and $\underline{\Phi}_{1,N+1}$ in the current macro time step at T_N and T_{N+1}. Note that the Lagrange basis polynomials from the macro

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

time window $[T_{N-2}, T_{N-1}]$ are used to calculate $\underline{\Phi}_{1,N}$ which results in a backward shift $h - -H = h + H$. Finally, the extrapolated data points are linearly interpolated to obtain continuous coupling values in the current macro time step. Thereto, the Lagrange basis polynomials from the current macro time window $[T_N, T_{N+1}]$ are applied to calculate $\underline{\tilde{u}}_1^{(k)}(h)$ which entails a forward shift $h - H$. For the C^1-version of the extrapolated interpolation method, see Fig. 2, the time derivative of $\Phi_{1,N}$ and $\Phi_{1,N+1}$ is additionally taken into account and the input vector is smoothed by means of a cubic Hermite polynomial

$$\begin{aligned} \dot{\underline{\Phi}}_{1,N} &:= \sum_{i=0}^{k} \underline{y}_{2,N-1-i} \cdot \frac{d}{dt} L_{k-i}^{(k)}(h - -H)\Big|_{h=0} \\ \dot{\underline{\Phi}}_{1,N+1} &:= \sum_{i=0}^{k} \underline{y}_{2,N-i} \cdot \frac{d}{dt} L_{k-i}^{(k)}(h)\Big|_{h=H} \\ \underline{\tilde{u}}_1^{(k)}(h) &:= \underline{\Phi}_{1,N} \cdot H_{0,0}^{(1)}(h-H) + \underline{\Phi}_{1,N+1} \cdot H_{1,0}^{(1)}(h-H) \\ &\quad + \dot{\underline{\Phi}}_{1,N} \cdot H_{0,1}^{(1)}(h-H) + \dot{\underline{\Phi}}_{1,N+1} \cdot H_{1,1}^{(1)}(h-H) \qquad (C^1\text{-approx. in subsystem 1}). \end{aligned} \tag{12b}$$

Applying a sequential *Gauss-Seidel scheme* for the co-simulation as shown in Fig. 1, the second subsystem obtains coupling values $\underline{y}_{1,N-k+1}, \dots, \underline{y}_{1,N+1}$ from subsystem 1 including the updated value $\underline{y}_{1,N+1}$ at the current macro time point T_{N+1} (or at $h = H$ in terms of step sizes). Now, a polynomial interpolation $\underline{\Phi}_2$ of degree j can be used to approximate the input vectors of subsystem 2

$$\begin{aligned} \underline{\tilde{u}}_2^{(j)}(h) &= \underline{\Phi}_2\left([-(j-1)\cdot H, \dots, -H, 0, H], \left[\underline{y}_{1,N+1-j}, \dots, \underline{y}_{1,N-1}, \underline{y}_{1,N}, \underline{y}_{1,N+1}\right]; h\right) \\ &= \sum_{i=0}^{j} \underline{y}_{1,N+1-i} \cdot L_{j-i}^{(j)}(h-H) \qquad \text{(Classical polynomial interpolation in subsystem 2).} \end{aligned} \tag{13}$$

Note that the Lagrange basis polynomials are shifted into the current macro time window $[T_N, T_{N+1}]$ in order to involve the coupling values at T_{N+1}. Using a parallel *Jacobi scheme*, the extrapolated interpolation method, see Eq. (12a) or Eq. (12b), is applied in both subsystems which results in a corresponding equation for subsystem 2:

$$\begin{aligned} \underline{\Phi}_{2,N} &:= \underline{\Phi}_2\left([-(j+1)\cdot H, \dots, -2H, -H], \left[\underline{y}_{1,N-(j+1)}, \dots, \underline{y}_{1,N-2}, \underline{y}_{1,N-1}\right]; h = 0\right) \\ &= \sum_{i=0}^{j} \underline{y}_{1,N-1-i} \cdot L_{j-i}^{(j)}(h - -H)\Big|_{h=0} \qquad \text{(Classical polynomial extrapolation at } T_N) \\ \underline{\Phi}_{2,N+1} &:= \underline{\Phi}_2\left([-j\cdot H, \dots, -H, 0], \left[\underline{y}_{1,N-j}, \dots, \underline{y}_{1,N-1}, \underline{y}_{1,N}\right]; h = H\right) \\ &= \sum_{i=0}^{j} \underline{y}_{1,N-i} \cdot L_{j-i}^{(j)}(h)\Big|_{h=H} \qquad \text{(Classical polynomial extrapolation at } T_{N+1}) \\ \underline{\tilde{u}}_2^{(j)}(h) &:= \underline{\Phi}_{2,N} \cdot L_0^{(1)}(h-H) + \underline{\Phi}_{2,N+1} \cdot L_1^{(1)}(h-H) \qquad (C^0\text{-approximation in subsystem 2}). \end{aligned} \tag{14}$$

Comparing Eq. (12a) and (12b) it can be observed that the C^0- and the C^1-version of the extrapolated interpolation method are based on the same data points $\underline{y}_{2,N-i}$, $i = 0...k+1$. Hence, for the sake of a compact formulation only the C^0-version is considered in the following. The equations for the C^1-version can be derived in the same manner and will have a similar shape.

Let $q := \max(k, j)$ be the maximum polynomial degree used for generating $\Phi_{i,N}$ and $\Phi_{i,N+1}$ (i = 1, 2). For the *Gauss-Seidel* and the *Jacobi scheme* the approximated input vectors can be combined into a vector function

$$\begin{aligned} \underline{\tilde{u}}(h) = \sum_{i=0}^{q} \Big(& \underline{\underline{M}}_N \cdot \underline{\underline{\Lambda}}_{i,N}^{(k,j)}(H) \cdot \left[\underline{y}_{N-1-i} \cdot L_0^{(1)}(h-H) + \underline{y}_{N-i} \cdot L_1^{(1)}(h-H)\right] \\ & + \underline{\underline{M}}_{N+1} \cdot \underline{\underline{\Lambda}}_{i,N+1}^{(k,j)}(h) \cdot \underline{y}_{N+1-i} \Big) \end{aligned} \tag{15}$$

with the polynomial basis matrices

$$\underline{\underline{\Lambda}}_{i,N}^{(k,j)}(h) := \begin{pmatrix} \underline{\underline{I}} \cdot L_{k-i}^{(k)}(h) & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{I}} \cdot L_{j-i}^{(j)}(h) \end{pmatrix},$$

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

$$\underline{\underline{\Lambda}}^{(k,j)}_{i,N+1}(h) := \begin{pmatrix} \underline{\underline{I}} \cdot L^{(k)}_{k-i}(h-H) & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{I}} \cdot L^{(j)}_{j-i}(h-H) \end{pmatrix} \tag{16}$$

and the method-characterizing matrices $\underline{\underline{M}}_N$ and $\underline{\underline{M}}_{N+1}$

$$\underline{\underline{M}}_N, \underline{\underline{M}}_{N+1} := \begin{cases} \begin{pmatrix} \underline{\underline{0}} & \underline{\underline{I}} \\ \underline{\underline{I}} & \underline{\underline{0}} \end{pmatrix}, \begin{pmatrix} \underline{\underline{0}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} \end{pmatrix} & \text{for } \textit{Jacobi scheme}, \\ \begin{pmatrix} \underline{\underline{0}} & \underline{\underline{I}} \\ \underline{\underline{0}} & \underline{\underline{0}} \end{pmatrix}, \begin{pmatrix} \underline{\underline{0}} & \underline{\underline{0}} \\ \underline{\underline{I}} & \underline{\underline{0}} \end{pmatrix} & \text{for } \textit{Gauss-Seidel scheme} \text{ with interpolation of degree } j, \\ \begin{pmatrix} \underline{\underline{0}} & \underline{\underline{0}} \\ \underline{\underline{I}} & \underline{\underline{0}} \end{pmatrix}, \begin{pmatrix} \underline{\underline{0}} & \underline{\underline{I}} \\ \underline{\underline{0}} & \underline{\underline{0}} \end{pmatrix} & \text{for } \textit{Gauss-Seidel scheme} \text{ with interpolation of degree } k. \end{cases} \tag{17}$$

Substituting $\underline{\tilde{u}}$ from Eq. (15) into Eq. (7), yields for $h = H$ the state vector

$$\begin{aligned} \underline{z}_{N+1} = \exp\left[\underline{\underline{\tilde{A}}} \cdot H\right] \cdot \underline{z}_N + \sum_{i=0}^{q} \int_0^H \exp\left[\underline{\underline{\tilde{A}}} \cdot (H-\tau)\right] \cdot \underline{\underline{\tilde{B}}} \cdot \\ \left(\underline{\underline{M}}_N \cdot \underline{\underline{\Lambda}}^{(k,j)}_{i,N}(H) \cdot \left[\underline{y}_{N-1-i} \cdot L^{(1)}_0(\tau - H) + \underline{y}_{N-i} \cdot L^{(1)}_1(\tau - H))\right] + \underline{\underline{M}}_{N+1} \cdot \underline{\underline{\Lambda}}^{(k,j)}_{i,N+1}(\tau) \cdot \underline{y}_{N+1-i}\right) d\tau \end{aligned} \tag{18}$$

at the macro time point T_{N+1}. Inserting $\underline{z}_{N+1}$ from Eq. (18) into the output equation (8), yields the corresponding output vector $\underline{y}_{N+1}$. Collecting the state vector $\underline{z}_{N+1}$ and the output vector $\underline{y}_{N+1}$ into one single vector, the following system of recurrence equations is obtained from Eqs. (18) and (8)

$$\begin{aligned} \begin{pmatrix} \underline{z}_{N+1} \\ \underline{y}_{N+1} \end{pmatrix} = & \begin{pmatrix} \underline{\underline{0}} & \underline{\underline{K}}_{0,N+1} \\ \underline{\underline{0}} & \underline{\underline{\tilde{C}}} \cdot \underline{\underline{K}}_{0,N+1} + \underline{\underline{\tilde{D}}} \cdot \underline{\underline{M}}_{N+1} \cdot \underline{\underline{\Lambda}}^{(k,j)}_{0,N+1}(H) \end{pmatrix} \cdot \begin{pmatrix} \underline{z}_{N+1} \\ \underline{y}_{N+1} \end{pmatrix} \\ & + \begin{pmatrix} \exp\left[\underline{\underline{\tilde{A}}} \cdot H\right] & \underline{\underline{K}}_{0,N} \\ \underline{\underline{\tilde{C}}} \cdot \exp\left[\underline{\underline{\tilde{A}}} \cdot H\right] & \underline{\underline{\tilde{C}}} \cdot \underline{\underline{K}}_{0,N} + \underline{\underline{\tilde{D}}} \cdot \underline{\underline{M}}_N \cdot \underline{\underline{\Lambda}}^{(k,j)}_{0,N}(H) \cdot L^{(1)}_1(0) \end{pmatrix} \cdot \begin{pmatrix} \underline{z}_N \\ \underline{y}_N \end{pmatrix} \\ & + \begin{pmatrix} \underline{\underline{0}} & \underline{\underline{K}}_{0,N-1} \\ \underline{\underline{0}} & \underline{\underline{\tilde{C}}} \cdot \underline{\underline{K}}_{0,N-1} + \underline{\underline{\tilde{D}}} \cdot \underline{\underline{M}}_N \cdot \underline{\underline{\Lambda}}^{(k,j)}_{0,N}(H) \cdot L^{(1)}_0(0) \end{pmatrix} \cdot \begin{pmatrix} \underline{z}_{N-1} \\ \underline{y}_{N-1} \end{pmatrix} \\ & + \sum_{i=1}^{q} \left[\begin{pmatrix} \underline{\underline{0}} & \underline{\underline{K}}_{i,N+1} \\ \underline{\underline{0}} & \underline{\underline{\tilde{C}}} \cdot \underline{\underline{K}}_{i,N+1} + \underline{\underline{\tilde{D}}} \cdot \underline{\underline{M}}_{N+1} \cdot \underline{\underline{\Lambda}}^{(k,j)}_{i,N+1}(H) \end{pmatrix} \cdot \begin{pmatrix} \underline{z}_{N+1-i} \\ \underline{y}_{N+1-i} \end{pmatrix} \right. \\ & + \begin{pmatrix} \underline{\underline{0}} & \underline{\underline{K}}_{i,N} \\ \underline{\underline{0}} & \underline{\underline{\tilde{C}}} \cdot \underline{\underline{K}}_{i,N} + \underline{\underline{\tilde{D}}} \cdot \underline{\underline{M}}_N \cdot \underline{\underline{\Lambda}}^{(k,j)}_{i,N}(H) \cdot L^{(1)}_1(0) \end{pmatrix} \cdot \begin{pmatrix} \underline{z}_{N-i} \\ \underline{y}_{N-i} \end{pmatrix} \\ & \left. + \begin{pmatrix} \underline{\underline{0}} & \underline{\underline{K}}_{i,N-1} \\ \underline{\underline{0}} & \underline{\underline{\tilde{C}}} \cdot \underline{\underline{K}}_{i,N-1} + \underline{\underline{\tilde{D}}} \cdot \underline{\underline{M}}_N \cdot \underline{\underline{\Lambda}}^{(k,j)}_{i,N}(H) \cdot L^{(1)}_0(0) \end{pmatrix} \cdot \begin{pmatrix} \underline{z}_{N-1-i} \\ \underline{y}_{N-1-i} \end{pmatrix} \right] \end{aligned} \tag{19}$$

with $L^{(1)}_0(0) = 0$ and $L^{(1)}_1(0) = 1$ according to definition (9). The integral matrices $\underline{\underline{K}}_{i,N-1}$, $\underline{\underline{K}}_{i,N}$, and $\underline{\underline{K}}_{i,N+1}$ are given by

$$\begin{aligned} \underline{\underline{K}}_{i,N+1} &:= \int_0^H \exp\left[\underline{\underline{\tilde{A}}} \cdot (H-\tau)\right] \cdot \underline{\underline{\tilde{B}}} \cdot \underline{\underline{M}}_{N+1} \cdot \underline{\underline{\Lambda}}^{(k,j)}_{i,N+1}(\tau) d\tau, \\ \underline{\underline{K}}_{i,N} &:= \int_0^H \exp\left[\underline{\underline{\tilde{A}}} \cdot (H-\tau)\right] \cdot \underline{\underline{\tilde{B}}} \cdot \underline{\underline{M}}_N \cdot \underline{\underline{\Lambda}}^{(k,j)}_{i,N}(H) \cdot L^{(1)}_1(\tau) d\tau, \quad \text{and} \\ \underline{\underline{K}}_{i,N-1} &:= \int_0^H \exp\left[\underline{\underline{\tilde{A}}} \cdot (H-\tau)\right] \cdot \underline{\underline{\tilde{B}}} \cdot \underline{\underline{M}}_N \cdot \underline{\underline{\Lambda}}^{(k,j)}_{i,N}(H) \cdot L^{(1)}_0(\tau) d\tau. \end{aligned} \tag{20}$$

Note that the integrals can be calculated analytically. Rearranging terms in (19) yields a linear recurrence system of order $(p+1)$ for the unknown state and output vectors

$$\begin{pmatrix} \underline{z}_{N+1} \\ \underline{y}_{N+1} \end{pmatrix} = \left[\underline{\underline{I}} - \underline{\underline{\Omega}}_{N+1}\right]^{-1} \cdot \left[\underline{\underline{\Omega}}_N \cdot \begin{pmatrix} \underline{z}_N \\ \underline{y}_N \end{pmatrix} + \underline{\underline{\Omega}}_{N-1} \cdot \begin{pmatrix} \underline{z}_{N-1} \\ \underline{y}_{N-1} \end{pmatrix} + \dots + \underline{\underline{\Omega}}_{N-p} \cdot \begin{pmatrix} \underline{z}_{N-p} \\ \underline{y}_{N-p} \end{pmatrix}\right], \tag{21}$$

where $\underline{\underline{\Omega}}_i \in \mathbb{R}^{8\times 8}$ denote the coefficient matrices and $\underline{\underline{I}} \in \mathbb{R}^{8\times 8}$ the identity matrix. The order $(p+1)$ of the recurrence scheme depends on the polynomial degrees k, j and also on the applied co-simulation approach:

$$p := \begin{cases} \max(k+1, j+1) & \text{for } \textit{Jacobi scheme}, \\ \max(k+1, j-1) & \text{for } \textit{Gauss-Seidel scheme} \text{ with interpolation of degree } j, \\ \max(k-1, j+1) & \text{for } \textit{Gauss-Seidel scheme} \text{ with interpolation of degree } k. \end{cases} \tag{22}$$

Equation (21) is transferred into a recurrence system of order 1

$$\underbrace{\begin{pmatrix} \begin{pmatrix} \underline{z}_{N+1} \\ \underline{y}_{N+1} \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \underline{z}_{N+1-p} \\ \underline{y}_{N+1-p} \end{pmatrix} \end{pmatrix}}_{\underline{Y}_{N+1}} = \underbrace{\begin{pmatrix} \tilde{\underline{\underline{\Omega}}}_N & \tilde{\underline{\underline{\Omega}}}_{N-1} & \cdots & \tilde{\underline{\underline{\Omega}}}_{N-(p-1)} & \tilde{\underline{\underline{\Omega}}}_{N-p} \\ \underline{\underline{I}} & \underline{\underline{0}} & \cdots & \underline{\underline{0}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{I}} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \underline{\underline{0}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \cdots & \underline{\underline{0}} & \underline{\underline{I}} & \underline{\underline{0}} \end{pmatrix}}_{\underline{\underline{\Omega}}} \cdot \underbrace{\begin{pmatrix} \begin{pmatrix} \underline{z}_N \\ \underline{y}_N \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \underline{z}_{N-p} \\ \underline{y}_{N-p} \end{pmatrix} \end{pmatrix}}_{\underline{Y}_N} \tag{23}$$

where $\underline{\underline{\Omega}}$ denotes the companion matrix with the sub-matrices in the first row given by

$$\tilde{\underline{\underline{\Omega}}}_{N-i} := \left[\underline{\underline{I}} - \underline{\underline{\Omega}}_{N+1}\right]^{-1} \cdot \left[\underline{\underline{\Omega}}_{N-i}\right]. \tag{24}$$

The applied co-simulation approach is numerically stable if the spectral radius $\rho(\underline{\underline{\Omega}})$, which is calculated as the magnitude of the largest eigenvalue of $\underline{\underline{\Omega}}$, satisfies the condition

$$\rho\left(\underline{\underline{\Omega}}\right) := \left\|\lambda\left(\underline{\underline{\Omega}}\right)\right\|_\infty < 1. \tag{25}$$

The companion matrix $\underline{\underline{\Omega}}$ as well as the spectral radius ρ depend on the applied co-simulation approach (*Jacobi scheme*, *Gauss-Seidel scheme*), the modular modeling approach (*force/displacement*, *displacement/displacement coupling*), the mechanical parameters of the test model ($m_1, m_2, c_1, c_2, d_1, d_2, c_k$, and d_k) and also on the macro-step size H. Note that ρ explicitly indicates the numerical stability of the co-simulation approach. Since the time integration in the subsystems is carried out analytically, ρ does not depend on numerical subsystem solvers and their parameters.

For investigating the local error of the co-simulation in the current macro time step, the numerical solution calculated by the linear method (21) is compared to the analytical solution. Since (21) depends on the solution at previous macro time steps, it is assumed that these solutions are exact and do not contribute additional numerical errors into the current macro step. The local error in the macro time step $T_N \to T_{N+1}$ is defined as

$$\tilde{\underline{\varepsilon}}_{N+1}(H) = \left[\tilde{\underline{\underline{\Omega}}}_N \cdot \begin{pmatrix} \underline{z}(T_N) \\ \underline{y}(T_N) \end{pmatrix} + \dots + \tilde{\underline{\underline{\Omega}}}_{N-p} \cdot \begin{pmatrix} \underline{z}(T_{N-p}) \\ \underline{y}(T_{N-p}) \end{pmatrix} - \begin{pmatrix} \underline{z}(T_{N+1}) \\ \underline{y}(T_{N+1}) \end{pmatrix}\right] \tag{26}$$

where the exact solution vectors can be calculated by solving the strongly coupled problem (4) at $T_{N+1}, T_N, \dots, T_{N-p}$.

4 Simulation results

In the appendix the 3d-stability plots are shown for different co-simulation methods, see also Wiley online library. In the plots the spectral radius ρ is plotted over the coupling stiffness c_k, the coupling damping parameter d_k and the macro step size H. The fixed parameters of the model are $c_1 = 1E6$, $c_2 = 1E7$, $d_1 = 1$, $d_2 = 2$, $m_1 = 10$, $m_2 = 10$. The left points of the plots show the stable configurations $\rho < 1$, the right plots show the unstable configurations $\rho >= 1$. As can be seen in the plots, increasing c_k or d_k entails a numerical instability for the co-simulation. However, reducing the macro step size H always stabilized the method since all here-considered approaches are zero stable.

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

In Fig. 9 the stability for the sequential *Gauss-Seidel scheme* in combination with a *force/displacement coupling* approach can be found which is the most common coupling approach and implemented very often. In the left column the C^1-version of the extrapolated interpolation method (denoted as EXTRIPOLC1 in the following) is shown and compared to a classical Lagrange approximation method in the right column. Sub-figures (a), (b), (c), and (d) show the plots for increasing polynomial degree 0, 1, 2, and 3. Using linear polynomials (degree 1) instead of constant polynomials (degree 0) improves the stability since the instability regions decrease. Increasing the degree further, the instability region increases again, i.e. the co-simulation is destabilized. This is a typical behavior for the most coupling approaches and has been validated in technical applications, see e.g. Ref. [6]. Comparing both columns in Fig. 9, the stability of both methods is similar. This is a good result for the EXTRIPOLC1 method and a bit surprising at first glance since the extrapolated interpolation method depends on one additional previous macro time step compared to the Lagrange approach, see also Fig. 2, so that the overall polynomial degree is one degree higher and a higher instability could be assumed. However, in the recurrence equation (19) the extrapolated interpolation method is evaluated at the macro step size ($h = H$) in order to calculate the output vector $\underline{y}_{N+1}$. Due to the interpolation conditions (9) and (10) of the shifted Lagrange and Hermite basis polynomials the evaluation of (12b) at $h = H$ results in the vector $\underline{\Phi}_{1,N+1}$, which is simply the classical Lagrange method evaluated at $h = H$, see also Fig. 2 for a graphical representation. This explains the similar stability plots of both methods. On the other hand, the higher polynomial degree of the extrapolated interpolation method enters the calculation of the state vector $\underline{z}_{N+1}$ in the recurrence equation (19) which entails that the stability plots of both methods are not identical.

In Fig. 10 the corresponding stability plots are calculated for the parallel *Jacobi scheme*. Again, it can be stated that the instability region of the EXTRIPOLC1 method is only marginally larger than for the Lagrange approach. Comparing Fig. 10 to Fig. 9, the parallel *Jacobi scheme* is more unstable than the sequential *Gauss-Seidel scheme* which is related with the fact that the *Gauss-Seidel scheme* uses updated information of the current macro time step while the *Jacobi scheme* only applies extrapolated information from the previous macro steps.

In Fig. 11 the stability regions for the C^0-version of the extrapolated interpolation method (denoted as EXTRIPOLC0 in the following) are plotted. The left column shows plots for the *Gauss-Seidel scheme* which can directly be compared with Fig. 9, in the right column the plots are shown for the *Jacobi scheme* which can directly be compared with Fig. 10. Comparing the plots, the EXTRIPOLC0 method is slightly more stable than the EXTRIPOLC1 method, but also here the results resemble that of the classical Lagrange approximation approach.

Even though the *displacement/displacement coupling* approach is not as often implemented as the *force/displacement coupling* approach, the corresponding stability plots are shown in Figs. 12–14 for the sake of completeness. The differences between the approximation methods are more visible here, especially in Fig. 12(a) vs. (b). The Lagrange approach entails no unstable points while the EXTRIPOLC1 method shows a small instability region for large coupling stiffness c_k. Interestingly, comparing Fig. 12(g) with (h) the EXTRIPOLC1 method is even slightly more stable than the Lagrange approach. This also holds for the EXTRIPOLC0 method in Fig. 14(g).

The local error (26) of the approaches is plotted in Fig. 15 with respect to the macro step size H. In the first column the error diagrams for the EXTIPOLC1 method are shown, the second and third column show the plots for the EXTRIPOLC0 method and the Lagrange approach. In the first and second row of the figure matrix the results are plotted for the *force/displacement coupling* approach in combination with a *Gauss-Seidel scheme* and a *Jacobi scheme*, the third and fourth row show the diagrams for the *displacement/displacement coupling* approach in combination with a *Gauss-Seidel scheme* and a *Jacobi scheme*. Comparing the first with the third column, it can be seen that the EXTRIPOLC1 method entails the same error order as the Lagrange approach. On the other hand, comparing the second column with the third, the EXTRIPOLC0 method suffers from an order drop so that the error is bounded by order 3. The reduction of the local error order for the EXTRIPOLC0 method is caused by the linear smoothing approach (12a). Even if the degree of the underlying extrapolation polynomials is increased, the error order is limited by the degree of the linear smoothing polynomial. Of cause, such an order limit will also occur for the EXTRIPOLC1 method if the degree of the extrapolation polynomials is further increased ($k = j > 3$). Polynomial degrees larger than cubic are however rarely used in applications due the increased numerical instability.

Comparing the *Gauss-Seidel* with the *Jacobi scheme* (row 1 vs. row 2), one observes an error drop for the *Jacobi scheme* in all methods. This behavior is also related with the fact that the *Jacobi scheme* uses only information from previous macro time steps. For the Lagrange and the Hermite approach, the phenomenon is analyzed in detail in Ref. [7]. Using a *displacement/displacement coupling* approach, the general error drop can be avoided.[2]

[2] According to recent investigations of the global error of co-simulation methods the order reduction phenomenon in the local error does not affect the order of the global error, see Ref. [4], so that convergence of the co-simulation can still be guaranteed. However, the consideration of the error drop can be of interest in relation with an adaptive macro-step size control which is typically based on the local error of the co-simulation method, see e.g. Refs. [9, 29].

5 Conclusions

A continuous and differentiable approach for approximating the coupling variables in co-simulation methods is presented in the paper. The numerical stability and the local error of the method are investigated and compared to the results of a classical Lagrange approximation approach. It is shown that the stability behavior of the C^1-continuous approximation is similar to that of the Lagrange approach. Further, both methods entail the same order of the local error in combination with different numerical coupling schemes, especially using a sequential *Gauss-Seidel scheme* together with a *force/displacement coupling* technique which is very often utilized in technical applications. Hence, applying the C^1-continuous approximation instead of Lagrange approximation, at least a similar efficiency of the co-simulation can be stated. In technical applications the C^1-method may perform even better than the Lagrange approach, since discontinuities in the subsystem equations with respect to the coupling variables will not occur, i.e., the numerical solvers do not have to decrease their integration step size after each macro time step as it can occur with the Lagrange approach.

As a further result, applying a C^0-continuous approach for approximating the coupling variables, a decreased local error order is observed, so that smaller macro step sizes have to be used for obtaining the same local error as for the Lagrange approach. The stability behavior of the C^0-continuous method is however similar to that of the Lagrange approach and the C^1-continuous approximation method.

Summarizing the investigation, the C^1-continuous method should be chosen over the C^0-continuous method if a smooth approximation of the coupling variables is used for the co-simulation. The effort for implementing both methods in simulation programs is similar, especially, since the C^1-continuous method depends on the same sampling points. Further, it is not a limitation that the C^1-continuous method requires access to the time derivative of the underlying extrapolation polynomials, since these polynomials are available to the programmer in the coupling interface of the simulation programs and are not hidden within the subsystems itself[3].

For investigating the stability of the coupling approaches an analytic time integration is used in the subsystems since the applied test model is linear. In nonlinear application cases the overall stability and the error may also be affected by the numerical solvers in the subsystems. The influence of the solvers can however be minimized by using sufficiently small solver tolerances in the subsystems. It should further be remarked that the derived stability and error results only hold for a constant macro step size. Varying the macro step size during the co-simulation can entail additional instabilities. This aspect was however not scope of the current investigation.

Appendix A: Approximation schemes for C^0 and C^1-continuous extrapolated interpolation method based on constant and quadratic extrapolation polynomials

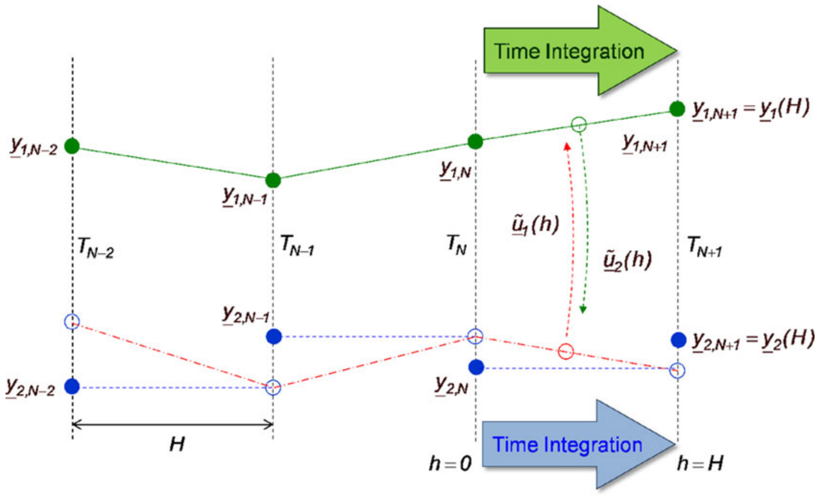


Fig. 5 C^0-continuous approximation using a constant polynomial for underlying extrapolation in the macro time steps. The scheme can be directly compared with Fig. 1.

[3] Remark: Subsystems generated with commercial simulation programs are mostly exported as black box, e.g. in a dynamic linked library (DLL), so that the model equations are hidden from the programmer or the coupling engineer. On the other hand, the coupling interface is normally open to the public, e.g. user force elements in multibody simulation programs.

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

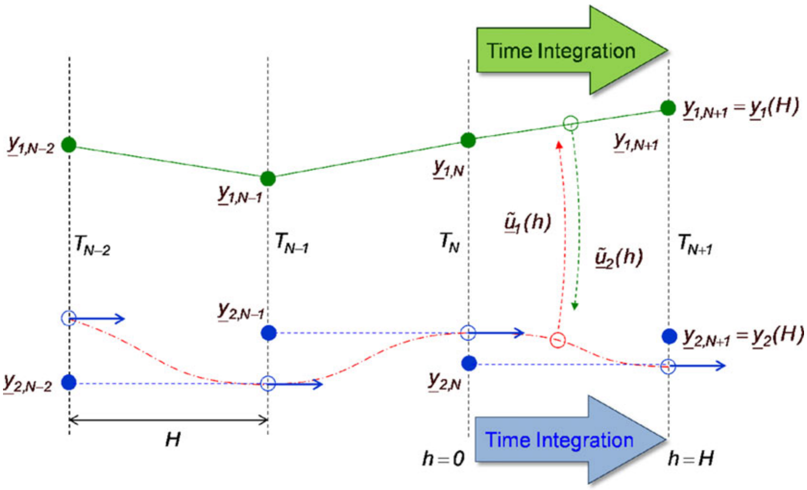


Fig. 6 C^1-continuous approximation using a constant polynomial for underlying extrapolation in the macro time steps. The scheme can be directly compared with Fig. 2.

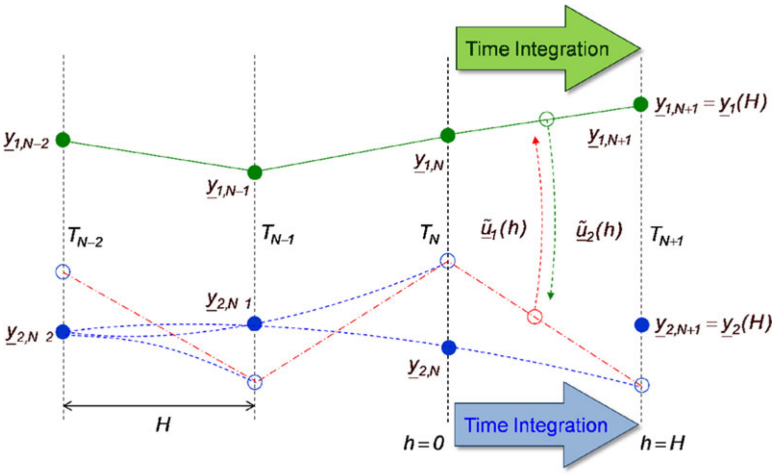


Fig. 7 C^0-continuous approximation using a quadratic polynomial for underlying extrapolation in the macro time steps. The scheme can be directly compared with Fig. 1.

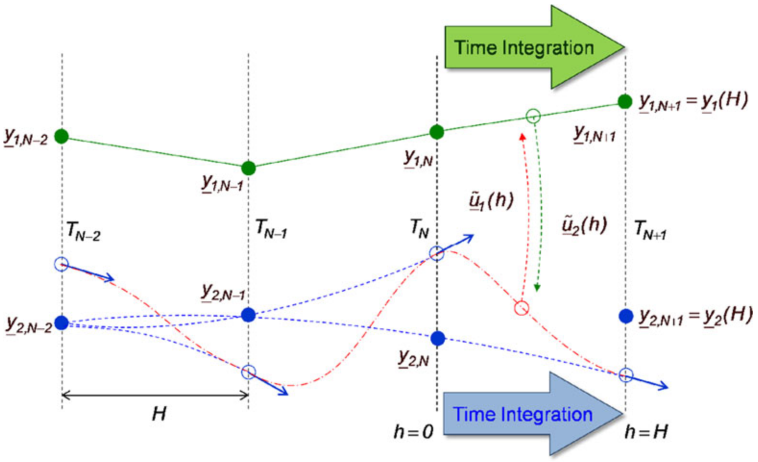


Fig. 8 C^1-continuous approximation using a quadratic polynomial for underlying extrapolation in the macro time steps. The scheme can be directly compared with Fig. 2.

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

Appendix B: Stability plots and local error diagrams

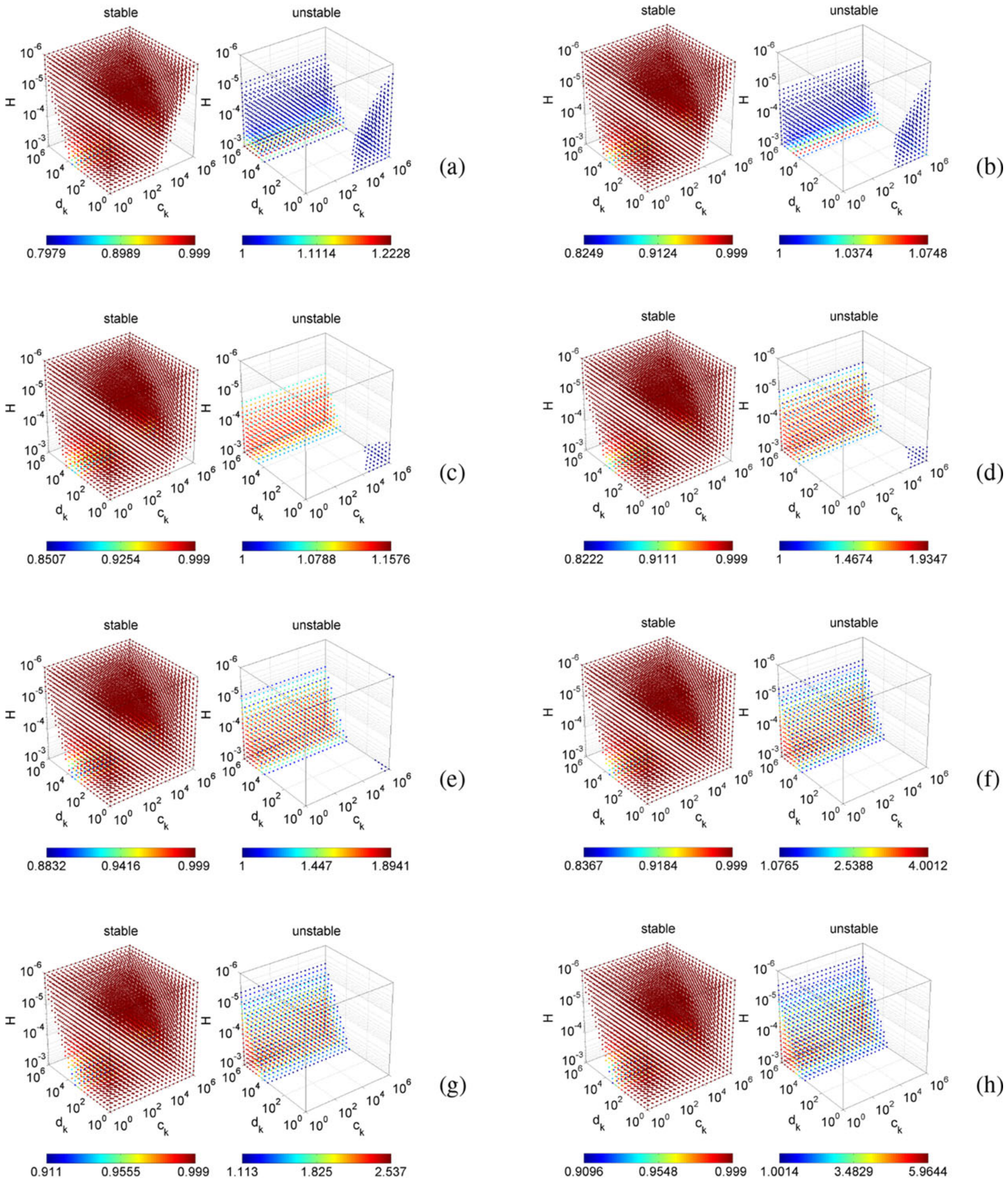


Fig. 9 3D-stability plots for a *force/displacement coupling* approach in combination with the *Gauss-Seidel scheme* using the C^1-version of the extrapolated interpolation (left column) and classical Lagrange polynomials (right column). The degrees of the approximation polynomials are: constant in (a) and (b), linear in (c) and (d), quadratic in (e) and (f), cubic in (g) and (h).

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

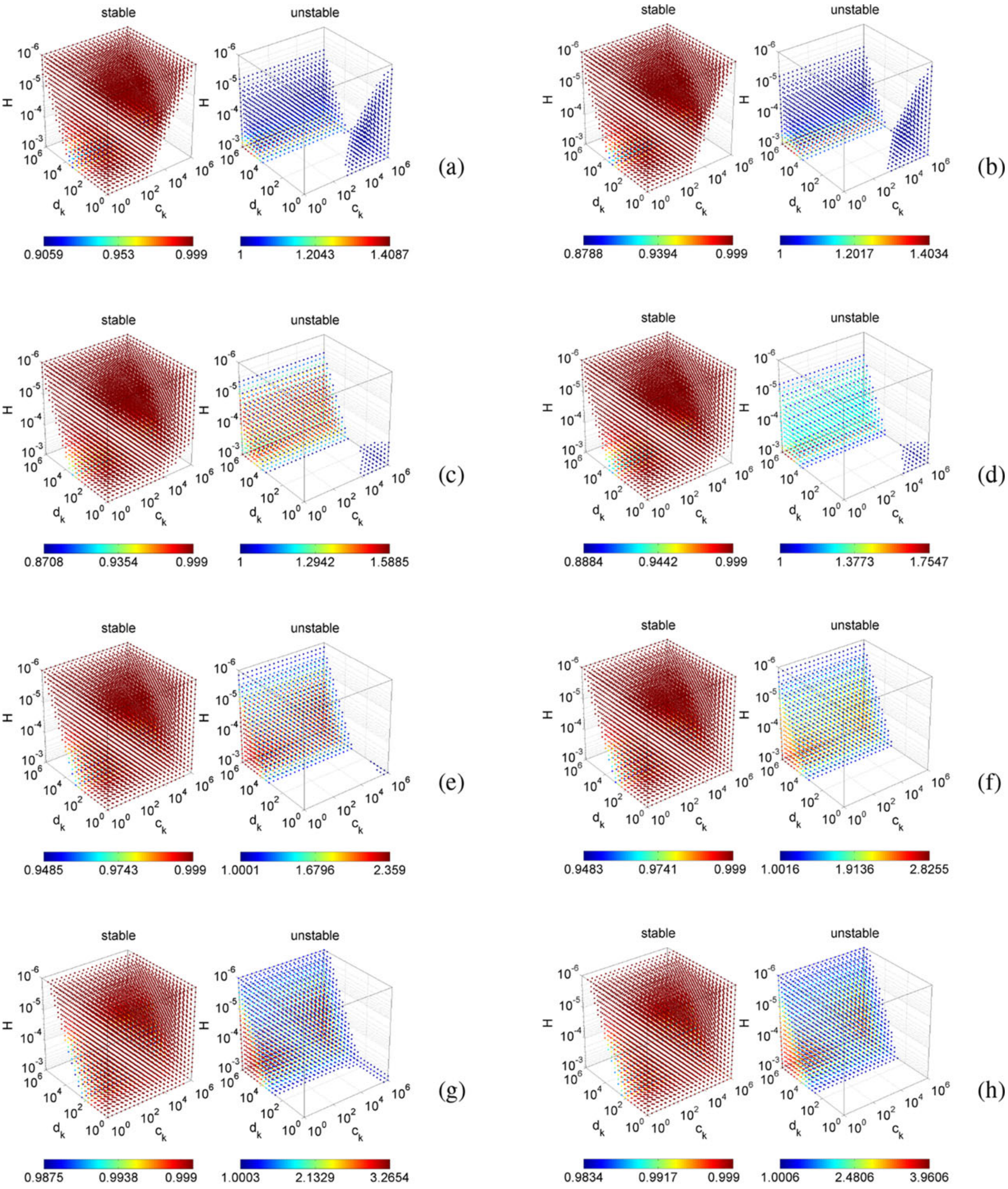


Fig. 10 Similar results as in Fig. 9. However, a *Jacobi scheme* is used for calculation.

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

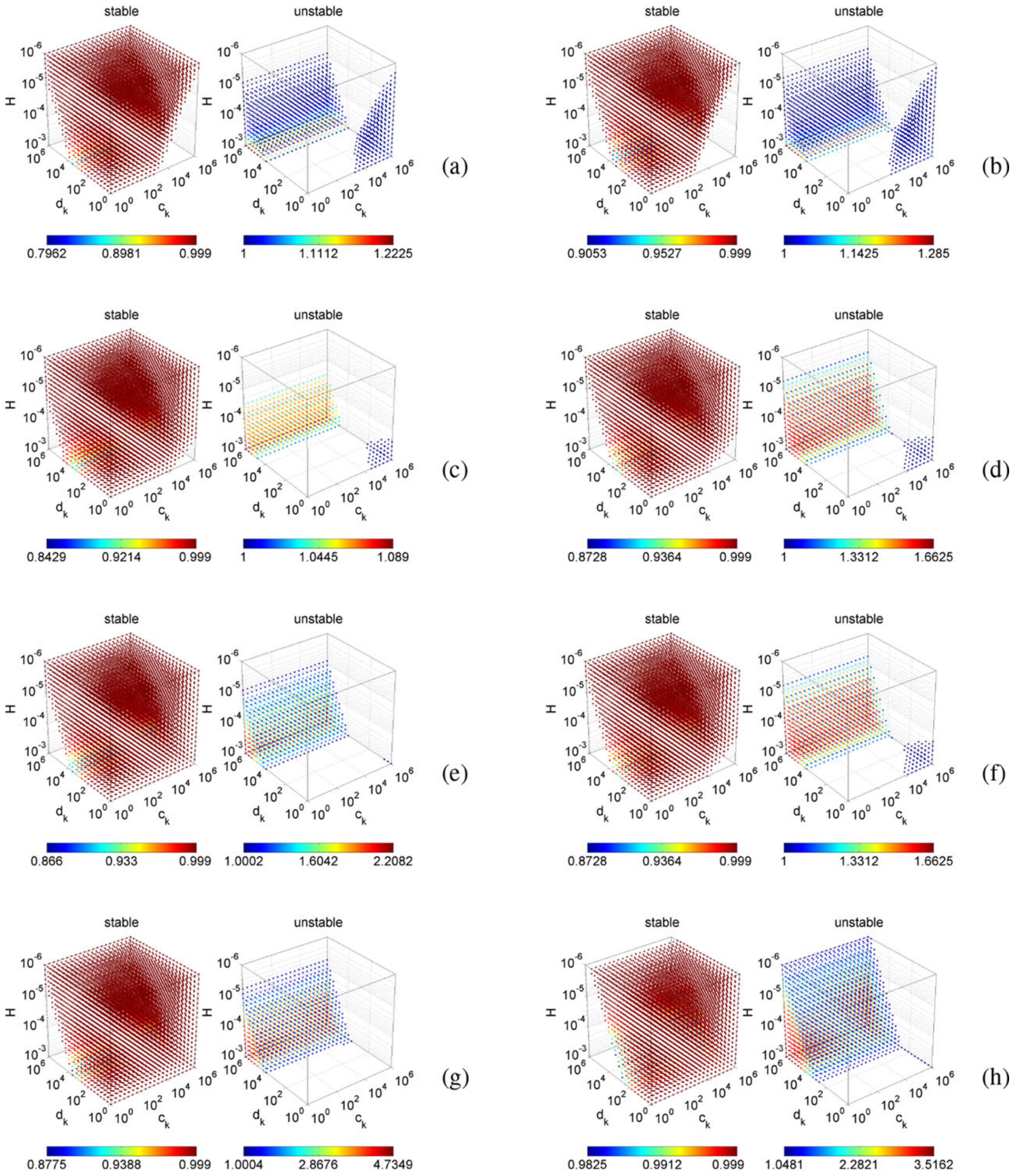


Fig. 11 Stability plots for the C^0-version of the extrapolated interpolation method in combination with a *Gauss-Seidel scheme* (left column) and a *Jacobi scheme* (right column), calculated for the *force/displacement coupling*.

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

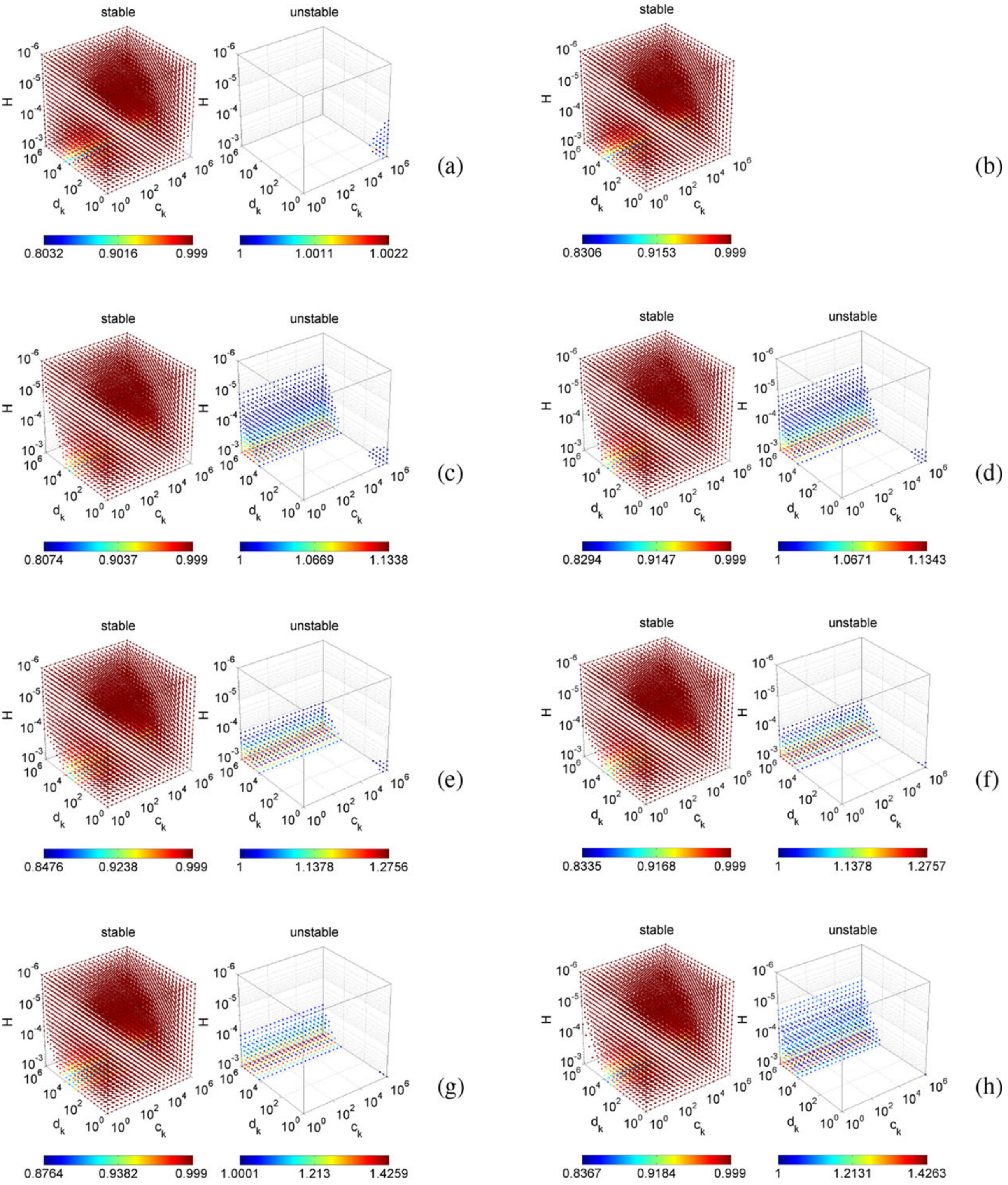


Fig. 12 3D-stability plots for a *displacement/displacement coupling* approach in combination with the *Gauss-Seidel scheme* using the C^1-continuous extrapolated interpolation (left column) and classical Lagrange polynomials (right column).

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

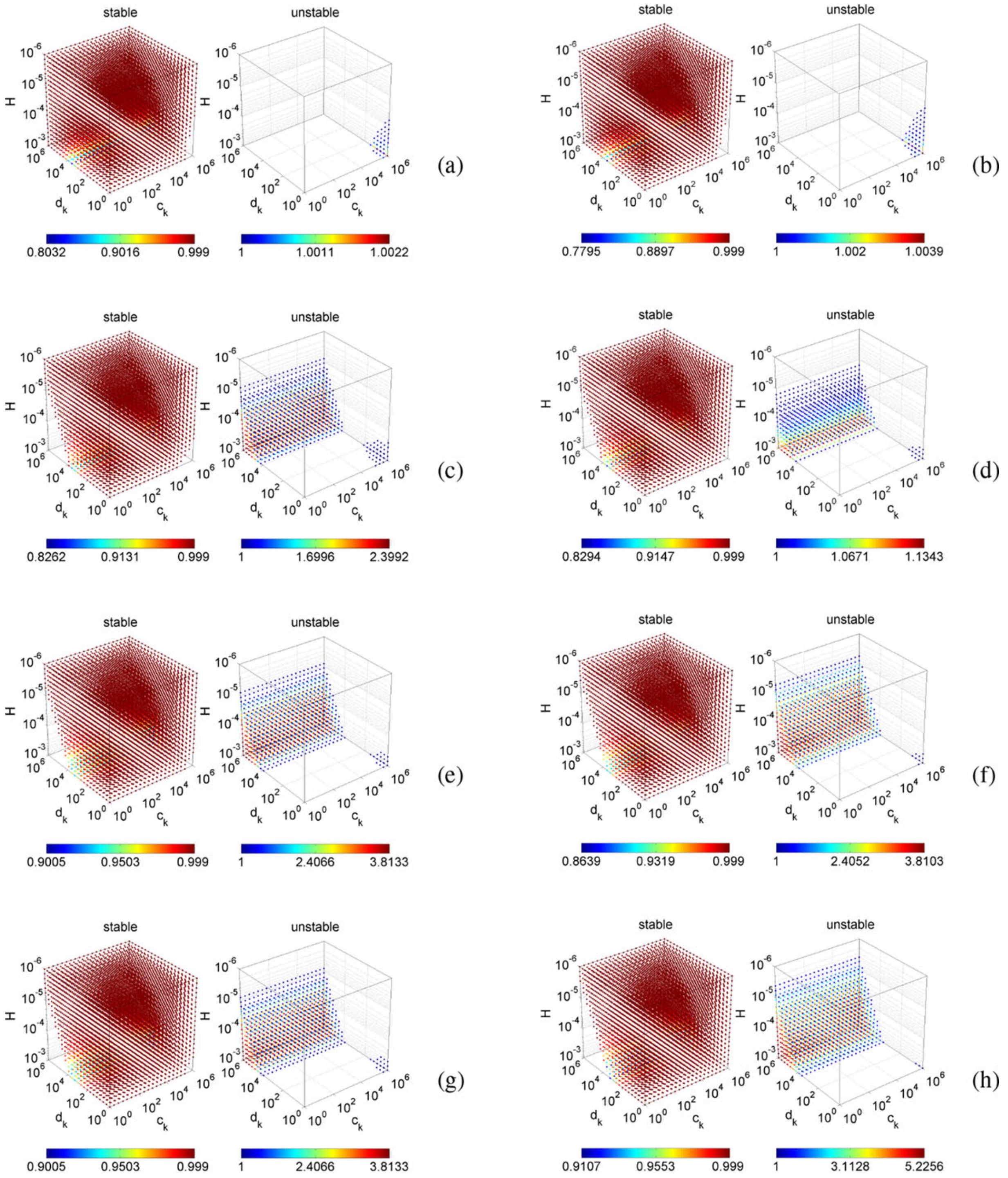


Fig. 13 Similar results as in left Fig. 12. However, a *Jacobi scheme* is used for calculation.

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

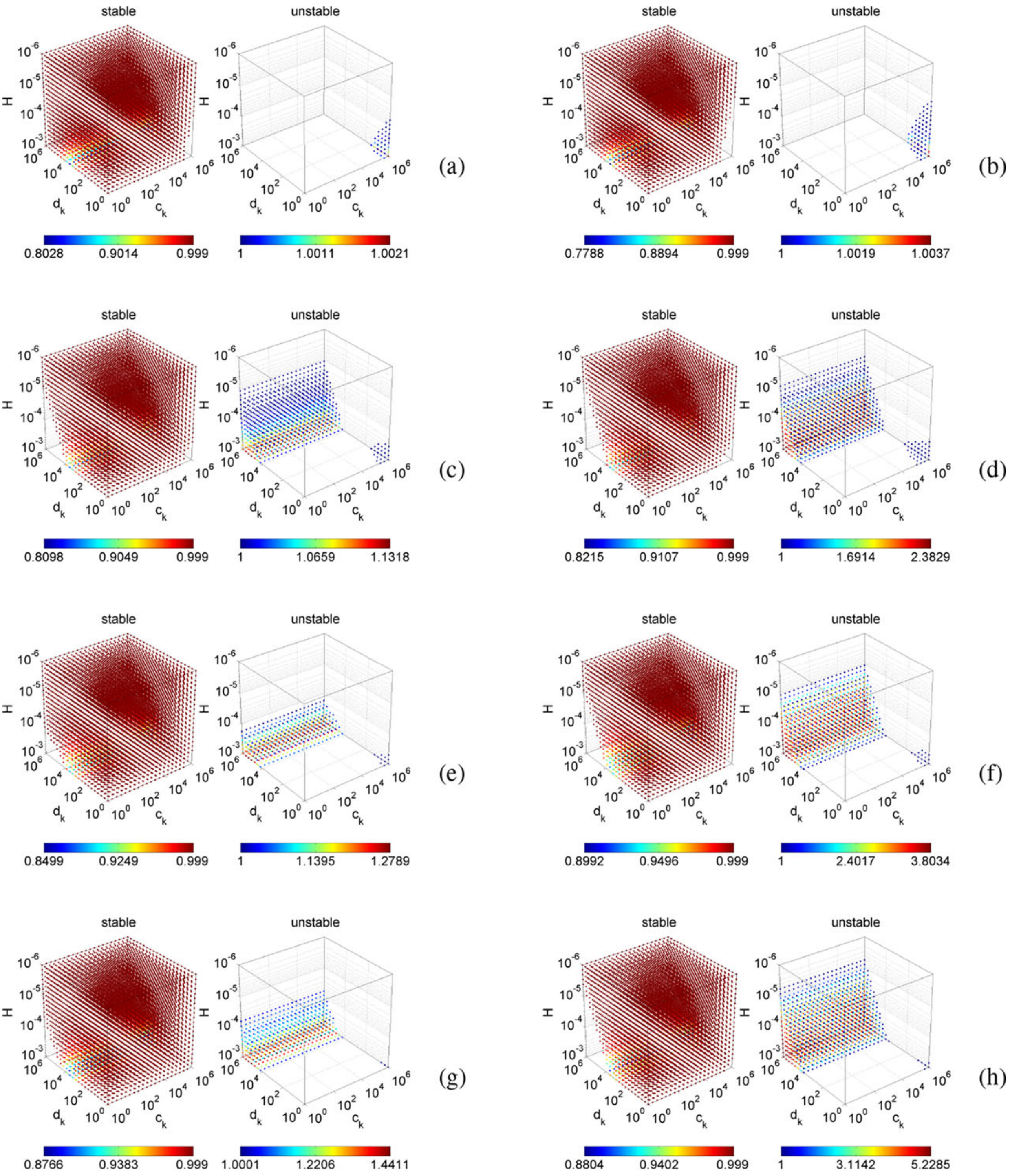


Fig. 14 Stability plots for the C^0-version of the extrapolated interpolation method in combination with a *Gauss-Seidel scheme* (left column) and a *Jacobi scheme* (right column), calculated for the *displacement/displacement coupling*.

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

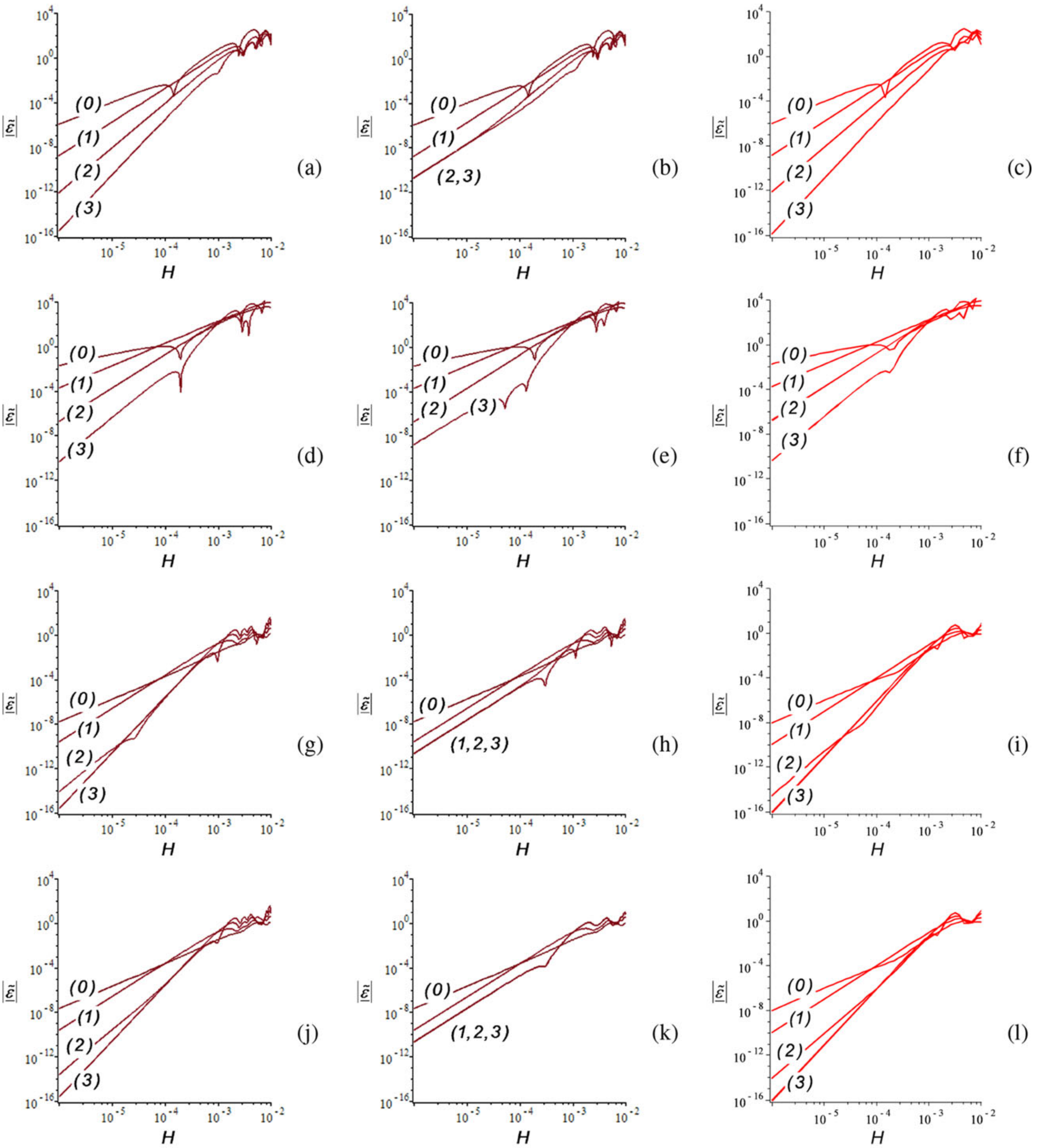


Fig. 15 Local error over the macro-step size for increasing polynomial degrees ($k = j = 0, 1, 2, 3$) and different coupling approaches: *Force/displacement coupling* + *Gauss-Seidel scheme* + EXTRIPOLC1 method (a), EXTRIPOLC0 method (b) and Lagrange approach (c); *Force/displacement coupling* + *Jacobi scheme* + EXTRIPOLC1 method (d), EXTRIPOLC0 method (e) and Lagrange approach (f); *displacement/displacement coupling* + *Gauss-Seidel scheme* + EXTRIPOLC1 method (g), EXTRIPOLC0 method (h) and Lagrange approach (i); *displacement/displacement coupling* + *Jacobi scheme* + EXTRIPOLC1 method (j), EXTRIPOLC0 method (k) and Lagrange approach (l).

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

References

[1] J. Ambrosio, J. Pombo, F. Rauter, and M. Pereira, A memory based communication in the co-simulation of multibody and finite element codes for pantograph-catenary interaction simulation, Computational Methods in Applied Sciences **12**, 231–252 (2008).

[2] M. Arnold, Multi-rate time integration for large scale multibody system models, in: IUTAM Symposium on Multiscale Problems in Multibody System Contacts, edited by P. Eberhard (Springer, Dordrecht, 2007), pp. 1–10.

[3] M. Arnold, Stability of sequential modular time integration methods for coupled multibody system models, Journal of Computational and Nonlinear Dynamics **5**, 031003 (2010).

[4] M. Arnold, C. Clauss, and T. Schierz, Error analysis and error estimates for co-simulation in FMI for Model Exchange and Co-Simulation V2.0, Archive of Mechanical Engineering **60**, 75–94 (2013).

[5] M. Arnold and M. Günther, Preconditioned dynamic iteration for coupled differential–algebraic systems, BIT Numerical Mathematics **41**, 1–25 (2001).

[6] M. Busch, Entwicklung einer SIMPACK-Modelica/Dymola Schnittstelle, Tech. Rep. DLR-IB-515-07-02, DLR Deutsches Zentrum für Luft- und Raumfahrt e.V, Oberpfaffenhofen, 2007.

[7] M. Busch, Zur effizienten Kopplung von Simulationsprogrammen (On the efficient coupling of simulation codes), PhD thesis, University of Kassel, Kassel, 2012, ISBN-13: 978-3862192960.

[8] M. Busch and B. Schweizer, Numerical stability and accuracy of different co-simulation techniques: Analytical investigations based on a 2-dof test model, in: Proceedings of The 1st Joint International Conference on Multibody System Dynamics, (Lappeenranta, 2010), pp. 1–13.

[9] M. Busch and B. Schweizer, An explicit approach for controlling the macro-step size of co-simulation methods, in: Proc. of 7th European Nonlinear Dynamics Conference, (Rome, 2011).

[10] M. Busch and B. Schweizer, Stability of co-simulation methods using hermite and lagrange approximation techniques, in: Proc. of ECCOMAS Thematic Conference on Multibody Dynamics (Brussels, 2011).

[11] M. Busch and B. Schweizer, Co-simulation of multibody and finite-element systems: An efficient and robust semi-implicit coupling approach, Archives of Applied Mechanics **82**, 723–741 (2012).

[12] G. Dahlquist, Convergence and stability in the numerical integration of ordinary differential equations, Math. Scand. **4**, 33–53 (1956).

[13] G. de Micheli and A. Sangiovanni-Vincentelli, Characterization of integration algorithms for the timing analysis of MOS VLSI circuits, Circuit Theory and Applications **10**, 299–309 (1982).

[14] S. Dronka and J. Rauh, Co-simulation-interface for user-force-elements, in: Proceedings of SIMPACK user meeting (Baden-Baden, 2006).

[15] F. Fleissner and P. Eberhard, A Co-Simulation Approach for the 3D Dynamic Simulation of Vehicles Considering Sloshing in Cargo and Fuel Tanks, PAMM **9**, 133–134 (2009).

[16] M. Friedrich and H. Ulbrich, A parallel co-simulation for mechatronic systems, in: Proceedings of the 1st Joint International Conference on Multibody System Dynamics (Lappeenranta, 2010), pp. 1–10.

[17] D. Fritzson, J. Stahl, and I. Nakhimovski, Transmission line co-simulation of rolling bearing applications, in: Proceedings of the 48th Conference on Simulation and Modelling (SIMS'07), (Göteborg, Sweden, 2010), pp. 24–39.

[18] C. W. Gear and D. R. Wells, Multirate linear multistep methods, BIT Numerical Mathematics **24**, 484–502 (1984).

[19] F. González, M. González, and J. Cuadrado, Weak coupling of multibody dynamics and block diagram simulation tools, in: Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (San Diego, 2009).

[20] B. Gu and H. Asada, Co-simulation of algebraically coupled dynamic subsystems without disclosure of proprietary subsystem models, Journal of Dynamic Systems, Measurement and Control **126**, 1–13 (2004).

[21] M. Günther, A. Kværnø, and P. Rentrop, Multirate partitioned runge-kutta methods, BIT Numerical Mathematics **41**, 504–514 (2001).

[22] M. Günther and P. Rentrop, Multirate row methods and latency of electric circuits, Applied Numerical Mathematics **13**, 83–102 (1999).

[23] S. Knorr, Multirateverfahren in der Co-Simulation, Master's thesis, Universität Ulm, 2002.

[24] R. Kübler and W. Schiehlen, Two methods of simulator coupling, Mathematical and Computer Modelling of Dynamical Systems **6**, 93–113 (2000).

[25] E. Lelarasmee, A. Ruehli, and A. Sangiovanni-Vincentelli, The waveform relaxation method for time domain analysis of large scale integrated circuits, IEEE Trans. on CAD of IC and Syst. **1**, 131–145 (1982).

[26] L. Li, R. M. Seymour, and S. Baigent, Integrating biosystem models using waveform relaxation, Journal on Bioinformatics and Systems Biology **2008**, 308623 (2008).

[27] K. C. Park, J. C. Chiou, and J. D. Downer, Explicit-implicit staggered procedure for multibody dynamics analysis, Journal of Guidance, Control and Dynamics **13**, 562–570 (1990).

[28] A. Pichelkostner, Interfaces und Multirate-Cosimulation für Reifensimulationsmodelle, Master's thesis, Universität Stuttgart, 2004.

[29] T. Schierz, M. Arnold, and C. Clauß, Co-simulation with communication step size control in an FMI compatible master algorithm, in: Proceedings of the 9th International Modelica Conference (Munich, 2012).

[30] B. Schweizer and D. Lu, Semi-implicit co-simulation approach for solver coupling, Archive of Applied Mechanics **12**, 1739–1769 (2014).

[31] B. Schweizer and D. Lu, Stabilized index-2 co-simulation approach for solver coupling with algebraic constraints, Multibody System Dynamics **34**, 129–161 (2015).

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

[32] M. Valasek, Modeling, simulation and control of mechatronical systems, in: Simulation Techniques for Applied Dynamics, edited by M. Arnold and W. Schiehlen (Springer Wien NewYork, Udine, 2008), pp. 75–140.

[33] A. Verhoeven, E. J. W. ter Maten, R. M. M. Mattheij, and B. Tasić, BDF Stability analysis of the BDF slowest-first multirate methods, International Journal of Computer Mathematics **84**, 895–923 (2007).

[34] S. Wünsche, C. Clauß, P. Schwarz, and F. Winkler, Electro-thermal circuit simulation using simulator coupling, IEEE Trans. Very Large Scale Integration Systems **5**, 277–282 (1997).

© 2016 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim