Modular time integration

A framework for the analysis of time integration methods in co-simulation

Seminar at the Department of Automatic Control, Lund University
June 18, 2012, Lund, Sweden

Martin Arnold
martin.arnold@mathematik.uni-halle.de
Martin Luther University Halle-Wittenberg
NWF II – Institute of Mathematics
D – 06099 Halle (Saale), Germany

Joint work with **Tom Schierz** (Martin Luther University)

Outline

1. The Functional Mock-up Interface (FMI)
- FMI for Model Exchange and Co-Simulation v2.0

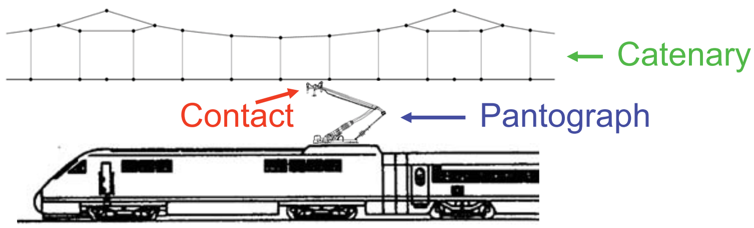
2. Co-Simulation: Local and global error
- Block oriented master-slave framework for coupled systems
- Order reduction (direct feed-through), Instability (algebraic loops)

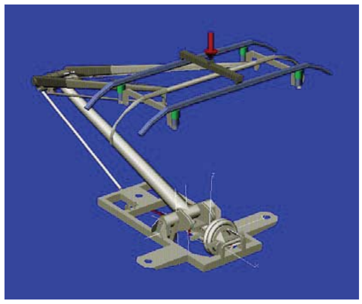
3. Communication step size control
- Local error estimates, Asymptotic analysis, Numerical tests

4. Stabilization using Jacobian matrices

Summary and Outlook

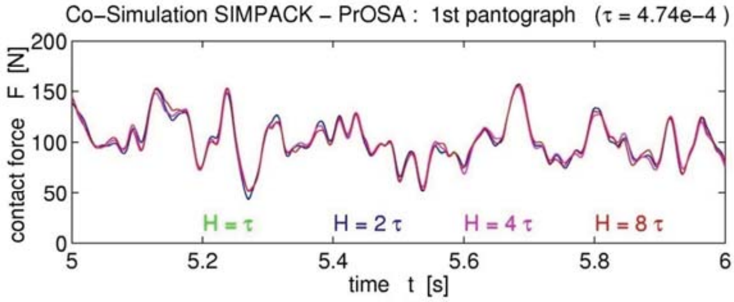
1. History of co-simulation: Pantograph / catenary





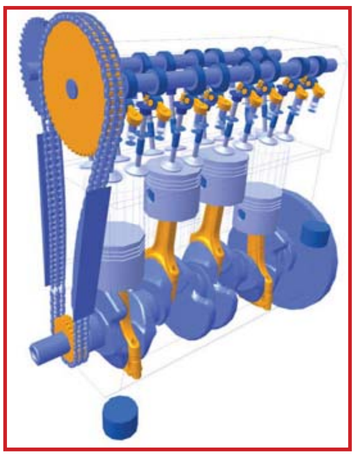
© A. Veitl (1999)

Pantograph: 56–DOF MBS, nonlinear
Catenary: Length 1200 m, $\tau < 4.75\text{e}{-4}\,\text{s}$

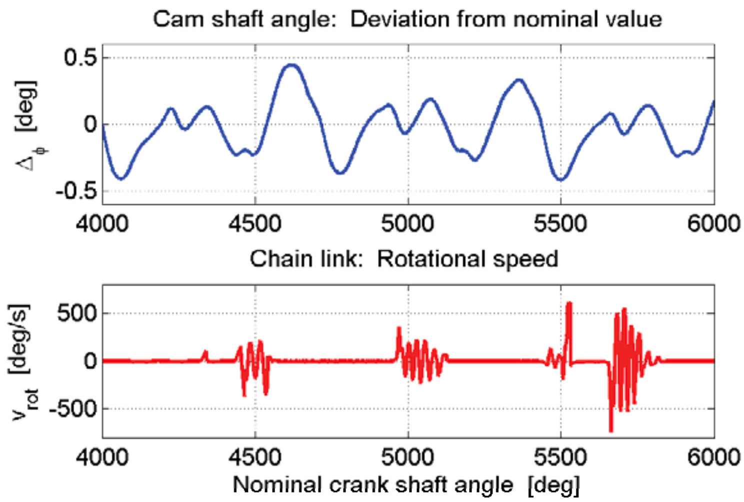


	SIMPACK	PrOSA	IPC
$H=\tau$	13496 s	601 s	15 s
$H=2\tau$	6677 s	590 s	10 s
$H=4\tau$	3373 s	578 s	11 s
$H=8\tau$	1680 s	573 s	10 s

Multi-rate time integration for multiscale problems



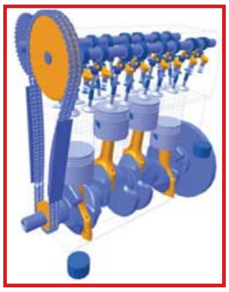
© M. Schittenhelm (2005)



Multi-scale problem: Different time stepsizes for differerent subsystems

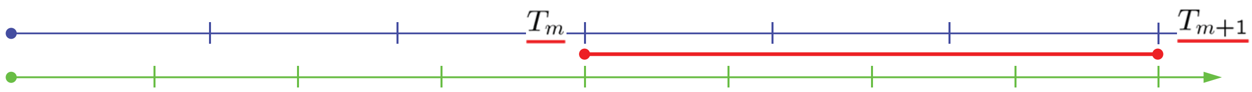
C.W. Gear, R.R. Wells: Multirate linear multistep methods. – BIT **24**(484-502)1984.
M. Günther, P. Rentrop: Multirate ROW methods and latency of electric circiuts. – Applied Numerical Mathematics **13**(83-102),1993.

Multi-rate time integration: Weak coupling



$$M_1(q_1)\ddot{q}_1 = f_1(q_1, \dot{q}_1, F_c(q_1, \dot{q}_1, q_2, \dot{q}_2))$$
$$M_2(q_2)\ddot{q}_2 = f_2(q_2, \dot{q}_2, F_c(q_1, \dot{q}_1, q_2, \dot{q}_2))$$

Core engine: High dimensional ODE or DAE model
Chain drive: High frequency oscillations
Coupling: Contact forces between chain links and wheels



Macro step $T_m \rightarrow T_{m+1} = T_m + H$ with macro stepsize H

- Time integration of subsystems by specialized solvers of multi-body dynamics (implicit DAE solvers based on high-order multistep methods, stepsize control, ...) with (different) micro-stepsizes h_1, h_2
- Data exchange between subsystems restricted to the discrete communication points T_m ("weak" coupling)

Multi-rate time integration: Engine / chain drive

- Subsystem 1: Core engine (DASSL), Subsystem 2: Chain drive (DOPRI5)
- Macro stepsize $H = 1.0\,\mu$s
- Higher order extrapolation and interpolation $F_c^{(m)}(t) = F_c(q_1(t), \bar{q}_2(t))$

$$\bar{q}_2(t) = q_2(T_m) + (t - T_m)\,\dot{q}_2(T_m) + \tfrac{1}{2}(t - T_m)^2\,\ddot{q}_2(T_m)$$

$$\bar{q}_1(t) = q_1(T_m) + (t - T_m)\,\frac{q_1(T_{m+1}) - q_1(T_m)}{T_{m+1} - T_m}$$

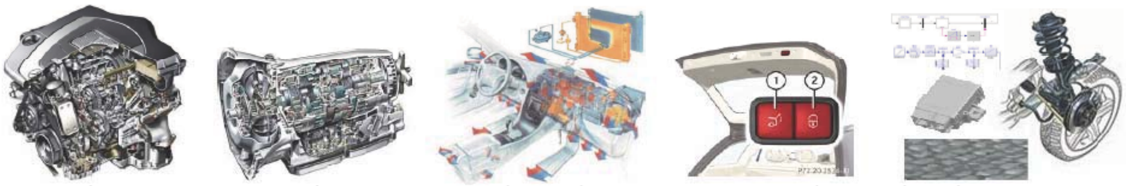
Numerical test

1: wheels, tensioner
2: chain links

	DASSL adapted	DASSL adapted+	LSODE	DOPRI5	Multi-rate integration
cpu-time [s]	18344.4	9585.5	3539.7	2487.9	748.1
# time steps	41287	43716	281494	60209	103384
# function evaluations	1705326	1325287	508891	361255	108013
# function evaluations (without Jacobian)	60606	62377	508891	361255	107272
# Jacobian evaluations	1232	946	0	0	19

MODELISAR: Functional Mock-up Interface

technical simulation models and embedded software



engine with ECU | gearbox with ECU | thermal systems | automated cargo door | chassis components, roadway, ECU (e.g. ESP)

co-simulation for functional mock-up

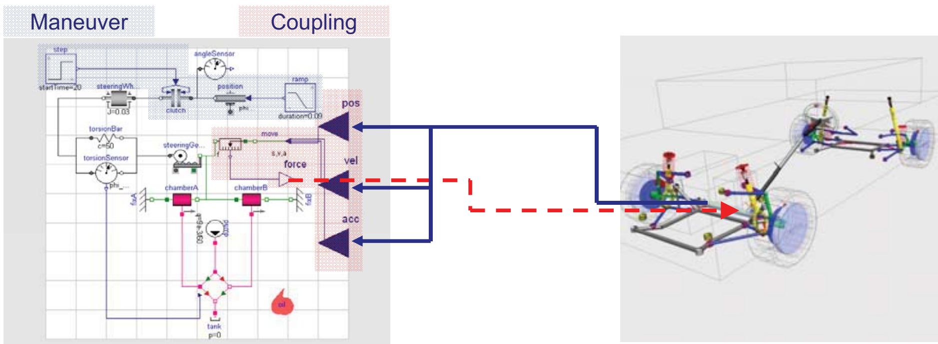
coupling of interacting technical systems and embedded software

- according to considered engineering task
- models from different engineering domains (geometry, mechanics, hydraulics, pneumatics, thermodynamics, electrics, electronics, cybernetics)

- numerically efficient and robust
- flexible and standardized

Case study: Car with servo-hydraulic steering

Servo-hydraulic steering



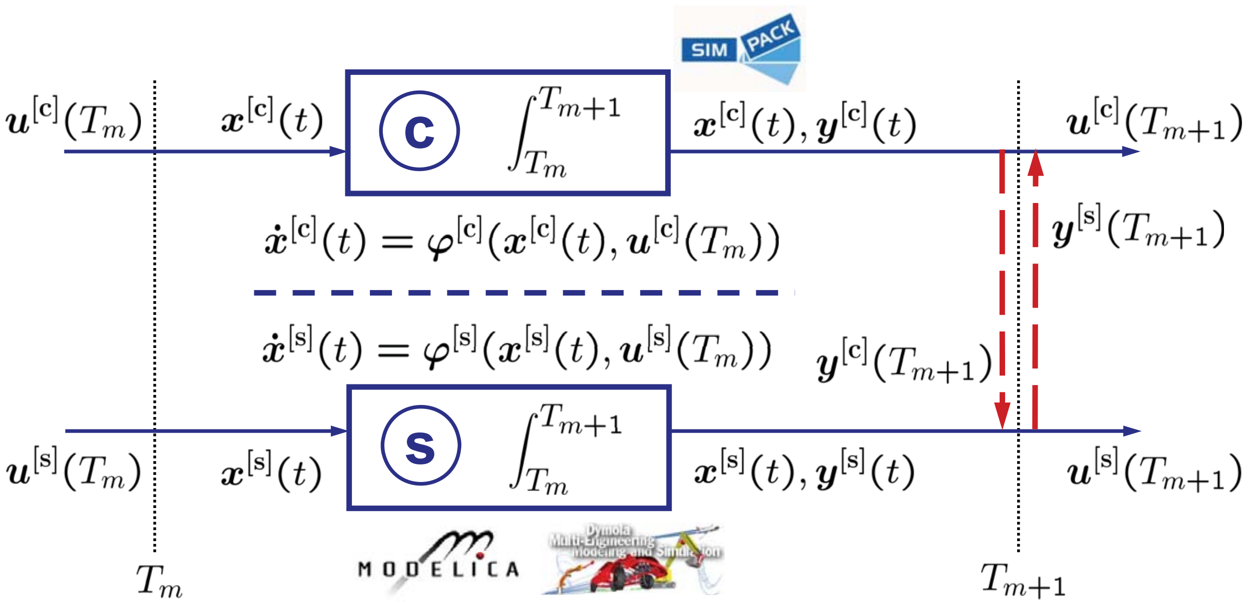
Detailed car model

$$\left.\begin{aligned} \dot{\boldsymbol{x}}^{[s]}(t) &= \varphi^{[s]}(\boldsymbol{x}^{[s]}(t), \boldsymbol{u}^{[s]}(t)) \\ \boldsymbol{y}^{[s]}(t) &= \gamma^{[s]}(\boldsymbol{x}^{[s]}(t)) \end{aligned}\right\} \qquad \left.\begin{aligned} \dot{\boldsymbol{x}}^{[c]}(t) &= \varphi^{[c]}(\boldsymbol{x}^{[c]}(t), \boldsymbol{u}^{[c]}(t)) \\ \boldsymbol{y}^{[c]}(t) &= \gamma^{[c]}(\boldsymbol{x}^{[c]}(t)) \end{aligned}\right\}$$

$$\begin{aligned} \boldsymbol{u}^{[s]}(t) &= \boldsymbol{y}^{[c]}(t) \quad \text{(rack excitation)} \\ \boldsymbol{u}^{[c]}(t) &= \boldsymbol{y}^{[s]}(t) \quad \text{(actuator force)} \end{aligned}$$

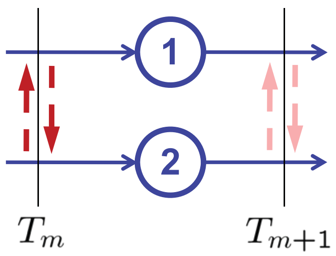
Co-Simulation: Modular time integration

Communication step / Macro step

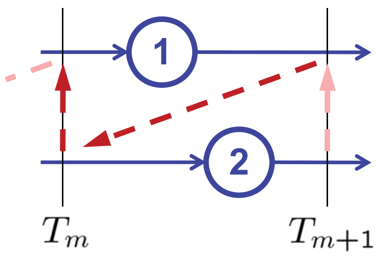


Co-Simulation: Algorithmic and numerical aspects

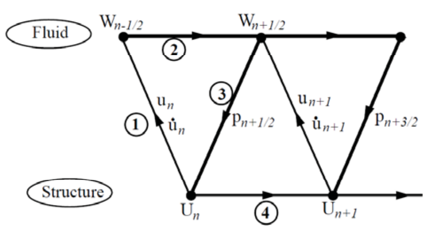
Parallel



Serial (staggered)



... and more



© Felippa, Park, Farhat (2001)

- Co-Simulation: Two, three, ... coupled subsystems
- Staggered algorithms exploit intermediate results in later stages
- Subsystems: Higher order methods, step size control, event handling, ...

Problem
- Data extrapolation (interpolation) to approximate coupling terms
- Additional error terms, Potential numerical instability

Benefits
- Customized solvers and time step sizes for subsystems

Modular time integration: State of the art vs. FMI

Co-Simulation

- Constant signal extrapolation
- Fixed communication step size
- Coupling terms are handled explicitly

Monolithic simulation tools

- Higher order time integration methods
- Variable time steps, Step size control
- Stiff ODEs, DAEs: Implicit methods

FMI for Model Exchange and Co-Simulation v2.0

- Higher order signal extrapolation: `fmiSetRealInputDerivatives`
- Communication step sizes of variable length: `fmiDoStep`
- Solver dumps to allow going back in time: `fmiGetFMUState`, `fmiSetFMUState`
- Evaluation of system Jacobians: `fmiGetPartialDerivatives`
- Capability flags
  - ✓ `canHandleVariableCommunicationStepSize`, `canGetAndSetFMUstate`
  - ✓ `providesPartialDerivativesOf_ ... _wrt_ ...`

http://www.fmi-standard.org/

2. Co-Simulation: Local and global error

FMI for Co-Simulation

Block-oriented master-slave framework

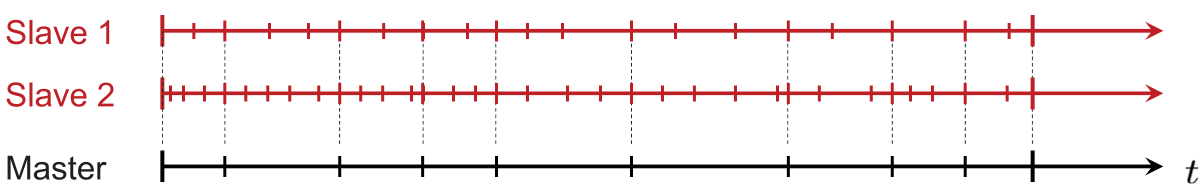
- $r \geq 2$ slave blocks

$$u_j(t) \longrightarrow \boxed{\dot{x}_j(t) = f_j(x_j(t), u_j(t))} \longrightarrow y_j(t) = g_j(x_j(t), u_j(t))$$

- Input-output coupling $u_j(t) = c_j(y_1(t), \ldots, y_{j-1}(t), y_{j+1}(t), \ldots, y_r(t))$
- Mathematical problem: Coupled system of differential equations

Co-Simulation

Data exchange master-slave at discrete communication points



Structure of the coupled system

$$\left.\begin{aligned} \dot{\boldsymbol{x}}_j(t) &= \boldsymbol{f}_j(\boldsymbol{x}_j(t), \boldsymbol{u}_j(t), \boldsymbol{u}_{\text{ex}}(t)) \\ \boldsymbol{y}_j(t) &= \boldsymbol{g}_j(\boldsymbol{x}_j(t), \boldsymbol{u}_j(t)) \end{aligned}\right\} \quad (j = 1, \ldots, r)$$

$$\boldsymbol{u}_j(t) = \boldsymbol{c}_j\Big(\boldsymbol{y}_1(t), \ldots, \boldsymbol{y}_{j-1}(t), \boldsymbol{y}_{j+1}(t), \ldots, \boldsymbol{y}_r(t)\Big), \quad (j = 1, \ldots, r)$$

Coupled system (DAE)

$$\left.\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{u}_{\text{ex}}(t)) \\ \boldsymbol{y}(t) &= \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \\ \boldsymbol{u}(t) &= \boldsymbol{c}(\boldsymbol{y}(t)) \end{aligned}\right\} \text{ with } \boldsymbol{x}(t) = \begin{pmatrix} \boldsymbol{x}_1(t) \\ \vdots \\ \boldsymbol{x}_r(t) \end{pmatrix}, \ \ldots$$

Special case Systems without direct feed-through: $\boldsymbol{y}_j(t) = \boldsymbol{g}_j(\boldsymbol{x}_j(t))$

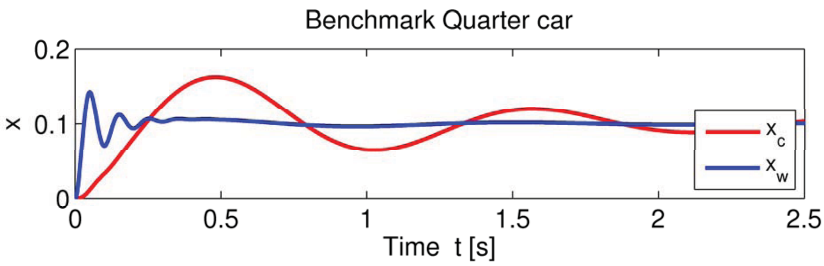
$$\text{ODE} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{u}_{\text{ex}}(t)) := \boldsymbol{f}\Big(\boldsymbol{x}(t), \boldsymbol{c}\big(\boldsymbol{g}(\boldsymbol{x}(t))\big), \boldsymbol{u}_{\text{ex}}(t)\Big)$$

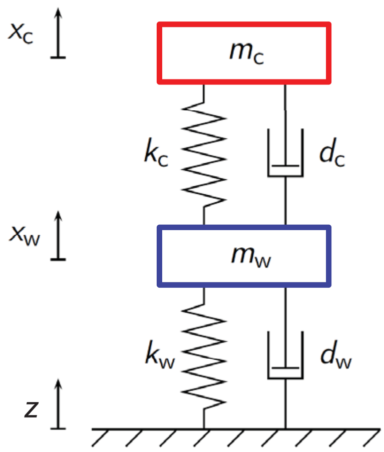
$$\text{Output equations} \quad \boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t)), \quad \boldsymbol{u}(t) = \boldsymbol{c}\big(\boldsymbol{g}(\boldsymbol{x}(t))\big)$$

Benchmark: Quarter car

Equations of motion

$$\begin{aligned} m_{\text{c}}\ddot{x}_{\text{c}} &= F_{\text{susp}}(x_{\text{c}}, \dot{x}_{\text{c}}, \underline{x_{\text{w}}}, \underline{\dot{x}_{\text{w}}}) \\ m_{\text{w}}\ddot{x}_{\text{w}} &= F_{\text{tyre}}(t, x_{\text{w}}, \dot{x}_{\text{w}}) - F_{\text{susp}}(\underline{x_{\text{c}}}, \underline{\dot{x}_{\text{c}}}, x_{\text{w}}, \dot{x}_{\text{w}}) \end{aligned}$$



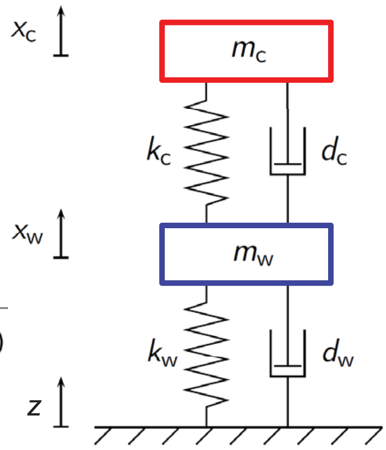


Suspension force $F_{\text{susp}}(\underline{x_{\text{c}}}, \underline{\dot{x}_{\text{c}}}, \underline{x_{\text{w}}}, \underline{\dot{x}_{\text{w}}}) = k_{\text{c}}(x_{\text{w}} - x_{\text{c}}) + d_{\text{c}}(\dot{x}_{\text{w}} - \dot{x}_{\text{c}})$

Tyre force $F_{\text{tyre}}(t, x_{\text{w}}, \dot{x}_{\text{w}}) = k_{\text{w}}(z(t) - x_{\text{w}}) + d_{\text{w}}(\dot{z}(t) - \dot{x}_{\text{w}})$
with $z(t) = 0$, $(t < 0)$, and $z(t) = 0.1$, $(t \geq 0)$.

Quarter car: Co-Simulation

$$m_\text{c}\ddot{x}_\text{c} = F_\text{susp}(x_\text{c},\dot{x}_\text{c},\underline{x_\text{w}},\underline{\dot{x}_\text{w}})$$
$$m_\text{w}\ddot{x}_\text{w} = F_\text{tyre}(t,x_\text{w},\dot{x}_\text{w}) - F_\text{susp}(\underline{x_\text{c}},\underline{\dot{x}_\text{c}},x_\text{w},\dot{x}_\text{w})$$



Displacement – Displacement coupling

$$\underline{\boldsymbol{u}_c} = \underline{\boldsymbol{y}_w},\quad m_\text{c}\ddot{x}_\text{c} = F_\text{susp}(x_\text{c},\dot{x}_\text{c},\underline{u_{\text{c},1}},\underline{u_{\text{c},2}})$$
$$\underline{\boldsymbol{y}_c} = \begin{pmatrix} x_c \\ \dot{x}_c \end{pmatrix}$$

$$\underline{\boldsymbol{u}_w} = \underline{\boldsymbol{y}_c},\quad m_\text{w}\ddot{x}_\text{w} = F_\text{tyre}(t,x_\text{w},\dot{x}_\text{w}) - F_\text{susp}(\underline{u_{\text{w},1}},\underline{u_{\text{w},2}},x_\text{w},\dot{x}_\text{w})$$
$$\underline{\boldsymbol{y}_w} = \begin{pmatrix} x_w \\ \dot{x}_w \end{pmatrix}$$

Force – Displacement coupling

$$\underline{u_c} = \underline{y_w},\quad m_\text{c}\ddot{x}_\text{c} = \underline{u_c},\quad \underline{\boldsymbol{y}_c} = \begin{pmatrix} x_c \\ \dot{x}_c \end{pmatrix}$$

$$\underline{\boldsymbol{u}_w} = \underline{\boldsymbol{y}_c},\quad m_\text{w}\ddot{x}_\text{w} = F_\text{tyre}(t,x_\text{w},\dot{x}_\text{w}) - F_\text{susp}(\underline{u_{\text{w},1}},\underline{u_{\text{w},2}},x_\text{w},\dot{x}_\text{w})$$
$$\underline{y_\text{w}} = F_\text{susp}(\underline{u_{\text{w},1}},\underline{u_{\text{w},2}},x_\text{w},\dot{x}_\text{w}) \qquad \Rightarrow\ \underline{y_\text{w}} = g_\text{w}(\ldots,\underline{\boldsymbol{u}_\text{w}})$$

Polynomial signal extrapolation: Local error

Communication step $T_m \to T_{m+1} = T_m + H$

Polynomial $\underline{\boldsymbol{\Psi}_j(t)} \approx \boldsymbol{u}_j(t)$ with $\boldsymbol{\Psi}_j(T_{m-i}) = \boldsymbol{u}_j(T_{m-i})$, $(\,i = 0,1,\ldots,k\,)$.

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{u}_\text{ex}(t)),\quad \boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t),\boldsymbol{u}(t)),\quad \boldsymbol{u}(t) = \boldsymbol{c}(\boldsymbol{y}(t))$$
$$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{f}(\hat{\boldsymbol{x}}(t),\underline{\boldsymbol{\Psi}(t)},\boldsymbol{u}_\text{ex}(t)),\quad \hat{\boldsymbol{y}}(t) = \boldsymbol{g}(\hat{\boldsymbol{x}}(t),\underline{\boldsymbol{\Psi}(t)}),\quad \hat{\boldsymbol{u}}(t) = \boldsymbol{c}(\hat{\boldsymbol{y}}(t))$$

Local error $\hat{\boldsymbol{x}}(T_m) := \boldsymbol{x}(T_m)$

$$\|\hat{\boldsymbol{x}}(T_{m+1}) - \boldsymbol{x}(T_{m+1})\| \le C_x\left(\mathrm{e}^{L_0(T_{m+1}-T_m)} - 1\right)\max_{t\in[T_m,T_{m+1}]}\|\underline{\boldsymbol{\Psi}(t)} - \boldsymbol{u}(t)\| = \mathcal{O}(\underline{H}\cdot\underline{H^{k+1}})$$
$$\|\hat{\boldsymbol{y}}(T_{m+1}) - \boldsymbol{y}(T_{m+1})\| \le C_y\left(\|\hat{\boldsymbol{x}}(T_{m+1}) - \boldsymbol{x}(T_{m+1})\| + \|\underline{\boldsymbol{\Psi}(T_{m+1})} - \boldsymbol{u}(T_{m+1})\|\right) = \mathcal{O}(\underline{H^{k+1}})$$
$$\|\hat{\boldsymbol{u}}(T_{m+1}) - \boldsymbol{u}(T_{m+1})\| \le C_u\|\hat{\boldsymbol{y}}(T_{m+1}) - \boldsymbol{y}(T_{m+1})\| = \mathcal{O}(\underline{H^{k+1}})$$

Order reduction in the local errors $\|\hat{\boldsymbol{y}}(T_{m+1}) - \boldsymbol{y}(T_{m+1})\|$, $\|\hat{\boldsymbol{u}}(T_{m+1}) - \boldsymbol{u}(T_{m+1})\|$
if $\big(\partial\boldsymbol{g}/\partial\boldsymbol{u}\big)(\boldsymbol{x},\boldsymbol{u}) \ne \boldsymbol{0}$ (direct feed-through) (Busch 2012)

Global error propagation and convergence

Coupled error propagation Global errors ϵ_m^{x}, ϵ_m^{u} (Kübler, Schiehlen 2000)

$$\|\epsilon_{m+1}^{x}\| \le \big(1+\mathcal{O}(H)\big)\|\epsilon_m^{x}\| + \mathcal{O}(H)\sum_{i=0}^{k}\|\epsilon_{m-i}^{u}\| + \mathcal{O}(\underline{H^{k+2}})$$

$$\epsilon_{m+1}^{u} = \sum_{i=0}^{k}\underline{J_m Z_{m-i}}\,\epsilon_{m-i}^{u} + \mathcal{O}(1)\|\epsilon_m^{x}\| + \mathcal{O}(H)\sum_{i=0}^{k}\|\epsilon_{m-i}^{u}\| + \mathcal{O}(\underline{H^{k+1}})$$

$$J_m := \frac{\partial c(g(x,\underline{u}))}{\partial u}\bigg|_{x=x(T_m),\,u=u(T_m)}, \quad Z_{m-i} := \frac{\partial \Psi}{\partial u_{m-i}}(T_m)$$

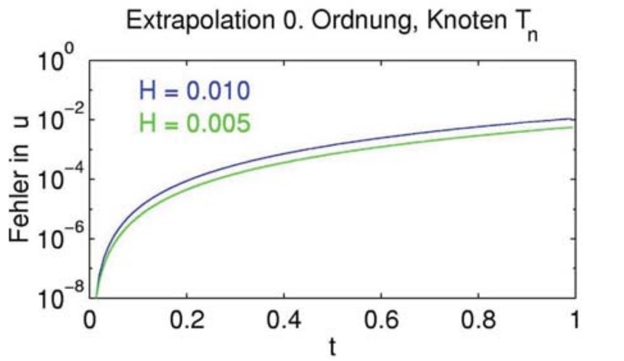
Contractivity condition to guarantee $\|\underline{J_m Z_{m-i}}\,\underline{J_{m-i} Z_{m-i-l}}\cdots\| \le C$ (zero stability)

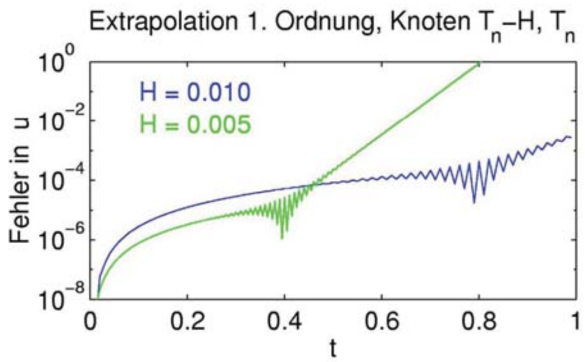
Exponential instability: DAE test problem

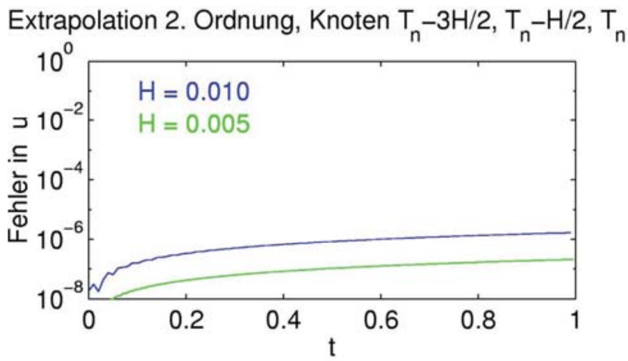
$$\dot y_1 = 4t^3, \quad 0 = -\tfrac{7}{2}y_1 + z_1 + u$$

$$\dot y_2 = 1, \quad 0 = \tfrac{2}{5}z_2 - u$$

$$0 = z_1 - z_2$$







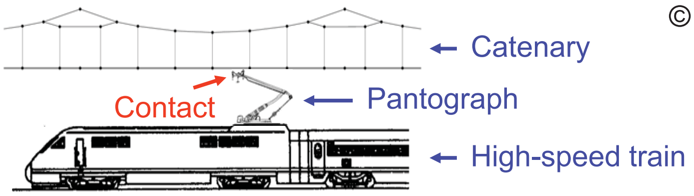
Exponential instability and stabilization: Example

Staggered algorithm $\alpha = \max \| (G_2 M_2^{-1} G_2^\top)^{-1} (G_1 M_1^{-1} G_1^\top) \| < 1$

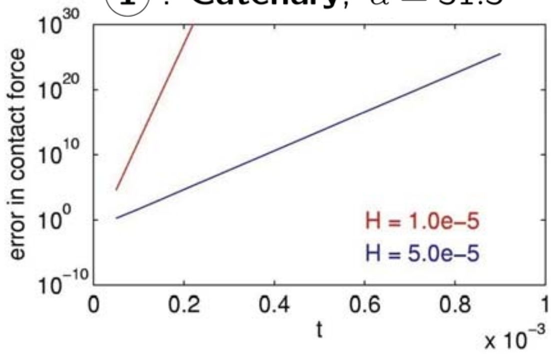
Benchmark

(A., Simeon 1998)

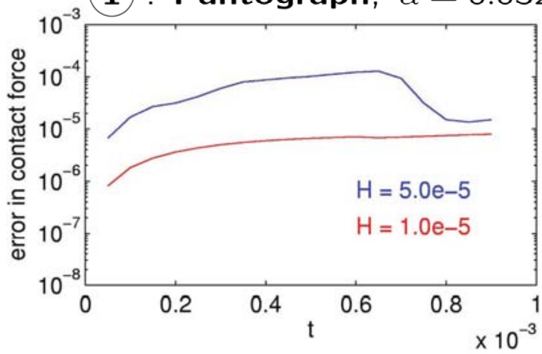
© A. Veitl (1999)



(1) : **Catenary**, $\alpha = 31.3$



(1) : **Pantograph**, $\alpha = 0.032$



Global error propagation: Structural Analysis

Coupled error propagation Global errors $\boldsymbol{\epsilon}_m^{\boldsymbol{x}}$, $\boldsymbol{\epsilon}_m^{\boldsymbol{u}}$ (Kübler, Schiehlen 2000)

$$\|\boldsymbol{\epsilon}_{m+1}^{\boldsymbol{x}}\| \le \big(1+\mathcal{O}(H)\big)\|\boldsymbol{\epsilon}_m^{\boldsymbol{x}}\| + \mathcal{O}(H)\sum_{i=0}^{k}\|\boldsymbol{\epsilon}_{m-i}^{\boldsymbol{u}}\| + \mathcal{O}(H^{k+2})$$

$$\boldsymbol{\epsilon}_{m+1}^{\boldsymbol{u}} = \sum_{i=0}^{k}\boldsymbol{J}_m\boldsymbol{Z}_{m-i}\,\boldsymbol{\epsilon}_{m-i}^{\boldsymbol{u}} + \mathcal{O}(1)\|\boldsymbol{\epsilon}_m^{\boldsymbol{x}}\| + \mathcal{O}(H)\sum_{i=0}^{k}\|\boldsymbol{\epsilon}_{m-i}^{\boldsymbol{u}}\| + \mathcal{O}(H^{k+1})$$

$$\boldsymbol{J}_m := \frac{\partial \boldsymbol{c}(\boldsymbol{g}(\boldsymbol{x},\boldsymbol{u}))}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}=\boldsymbol{x}(T_m),\,\boldsymbol{u}=\boldsymbol{u}(T_m)}, \quad \boldsymbol{Z}_{m-i} := \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{u}_{m-i}}(T_m)$$

- Contractivity condition to guarantee $\|\boldsymbol{J}_m\boldsymbol{Z}_{m-i}\,\boldsymbol{J}_{m-i}\boldsymbol{Z}_{m-i-l}\cdots\| \le C$ (zero stability)
- No direct feed-through in output equations $\Rightarrow$ $\boldsymbol{J}_m \equiv \boldsymbol{0}$ $\Rightarrow$ $\|\boldsymbol{\epsilon}_{m+1}^{\boldsymbol{x}}\| = \mathcal{O}(H^{k+1})$
- No algebraic loops: Convergence with $\|\boldsymbol{\epsilon}_{m+1}^{\boldsymbol{x}}\| = \mathcal{O}(H^{k+1})$ if $\boldsymbol{J}_m\boldsymbol{J}_{m-i_1}\cdots\boldsymbol{J}_{m-i_n} \equiv \boldsymbol{0}$

Structural analysis Structural non-zeros of Jacobian $\boldsymbol{J}_m$

Structural analysis to detect algebraic loops

$$\boxed{\boldsymbol{J}_m \boldsymbol{J}_{m-i_1} \cdots \boldsymbol{J}_{m-in} \equiv \boldsymbol{0}}$$

Displacement – Displacement coupling

$$\underline{\boldsymbol{u}_c} = \underline{\boldsymbol{y}_w} = \begin{pmatrix} x_w \\ \dot{x}_w \end{pmatrix}$$

$$\underline{\boldsymbol{u}_w} = \underline{\boldsymbol{y}_c} = \begin{pmatrix} x_c \\ \dot{x}_c \end{pmatrix}$$

$$\boldsymbol{J}_m = 0_{4\times 4} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Force – Displacement coupling

$$\underline{\boldsymbol{u}_c} = \underline{\boldsymbol{y}_w} = F_{\text{susp}}(\underline{u_{\text{W},1}}, \underline{u_{\text{W},2}}, x_{\text{W}}, \dot{x}_{\text{W}})$$

$$\underline{\boldsymbol{u}_w} = \underline{\boldsymbol{y}_c} = \begin{pmatrix} x_c \\ \dot{x}_c \end{pmatrix}$$

$$\boldsymbol{J}_m = \begin{pmatrix} 0 & * & * \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{J}_m \boldsymbol{J}_{m-i} = 0_{3\times 3}$$

Structural analysis to detect algebraic loops (II)

Coupling by constraints

$$\boldsymbol{M}_1 \underline{\ddot{\boldsymbol{q}}_1} = \boldsymbol{f}_1(\underline{\boldsymbol{q}_1}, \underline{\dot{\boldsymbol{q}}_1}) - \boldsymbol{G}_1^\top(\underline{\boldsymbol{q}_1}, \underline{\boldsymbol{q}_2})\boldsymbol{\lambda} \qquad \boldsymbol{M}_2 \underline{\ddot{\boldsymbol{q}}_2} = \boldsymbol{f}_2(\underline{\boldsymbol{q}_2}, \underline{\dot{\boldsymbol{q}}_2}) - \boldsymbol{G}_2^\top(\underline{\boldsymbol{q}_1}, \underline{\boldsymbol{q}_2})\boldsymbol{\lambda}$$

$$\boldsymbol{0} = \underline{\boldsymbol{g}(\boldsymbol{q}_1(t), \boldsymbol{q}_2(t))} \quad \overset{\mathrm{d}^2/\mathrm{d}t^2}{\Rightarrow} \quad \boldsymbol{0} = \underline{\boldsymbol{G}_1 \ddot{\boldsymbol{q}}_1 + \boldsymbol{G}_2 \ddot{\boldsymbol{q}}_2 + \boldsymbol{g}_{qq}(\boldsymbol{q}_1, \dot{\boldsymbol{q}}_1, \boldsymbol{q}_2, \dot{\boldsymbol{q}}_2)}$$

$$\boldsymbol{J}_m = \begin{pmatrix} 0 & 0 & * \\ 0 & 0 & * \\ * & * & 0 \end{pmatrix}, \quad \boldsymbol{J}_m \boldsymbol{J}_{m-i} = \begin{pmatrix} * & * & 0 \\ * & * & 0 \\ 0 & 0 & * \end{pmatrix}, \quad \ldots$$

Practical aspects

- Methods from graph theory (interpretation as adjacency matrix)
- Structural criterion: sufficient to exclude algebraic loops, not necessary
- Limitations: (Multiple) moving loads, e.g., railway bogie at track

3. Communication step size control

- Quantity of interest approach: Error bounds ATOL_i, RTOL_i for $\boldsymbol{y} = (y_i)_i$
- Errors in the subsystems should be negligible: $\text{TOL}_{\text{slave}} = 0.01\,\text{TOL}_{\text{master}}$

$$\texttt{err} := \left(\frac{1}{n_y} \sum_{i=1}^{n_y} \Big(\frac{\text{EST}_i}{\text{ATOL}_i + \text{RTOL}_i \cdot |y_i|} \Big)^2 \right)^{1/2} \quad \Rightarrow \quad H_{\text{opt}} = \alpha H_m \left(\frac{1}{\texttt{err}} \right)^{\frac{1}{q+1}}$$

- Communication step accepted ($\texttt{err} \le 1$) or rejected ($\texttt{err} > 1$)

Error estimate Richardson extrapolation (comparison of two numerical solutions)

Numerical solution $\bar{\boldsymbol{y}}(T_{m+2})$ after two (small) communication steps of size H

$$T_m \to T_{m+1} = T_m + H \to T_{m+2} = T_{m+1} + H = T_m + 2H$$

Numerical solution $\tilde{\boldsymbol{y}}(T_{m+2})$ after one large communication step of size $2H$

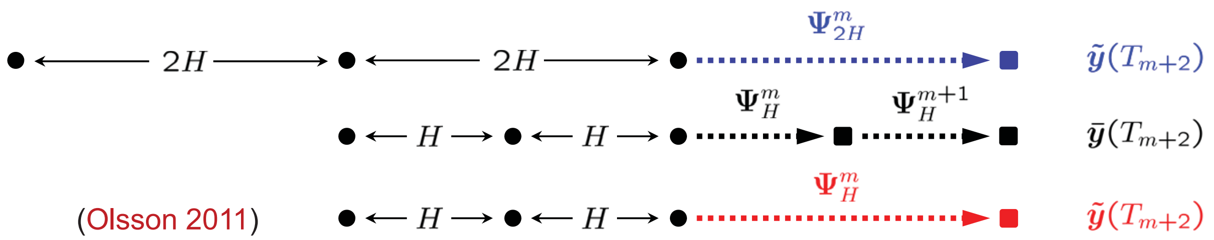
- with $\boldsymbol{\Psi}_{2H}(T_m - 2iH) = \boldsymbol{u}(T_m - 2iH)$, $(i = 0, 1, \ldots, k)$, (classical estimate)
- with $\boldsymbol{\Psi}_{H}(T_m - iH) = \boldsymbol{u}(T_m - iH)$, $(i = 0, 1, \ldots, k)$, (modified estimate)

Local error estimates: Asymptotic analysis

$$\text{EST}_{\text{Rich}} := \frac{\tilde{\boldsymbol{y}}(T_{m+2}) - \bar{\boldsymbol{y}}(T_{m+2})}{2^{k+1} - 1}$$

„Daraus lässt sich für die modulare Integration die Fehlerschätzung ... angeben.“ (Kübler 2000)

$$\text{EST}_{\text{Rich}} \doteq \text{le}_y^m + \frac{2^{k+1}(k+1)}{2^{k+1}-1} \Big(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\Big)_m \Big(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}}\Big)_m \Big(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\Big)_m \boldsymbol{u}^{(k+1)}(T_m) \cdot H^{k+1}$$



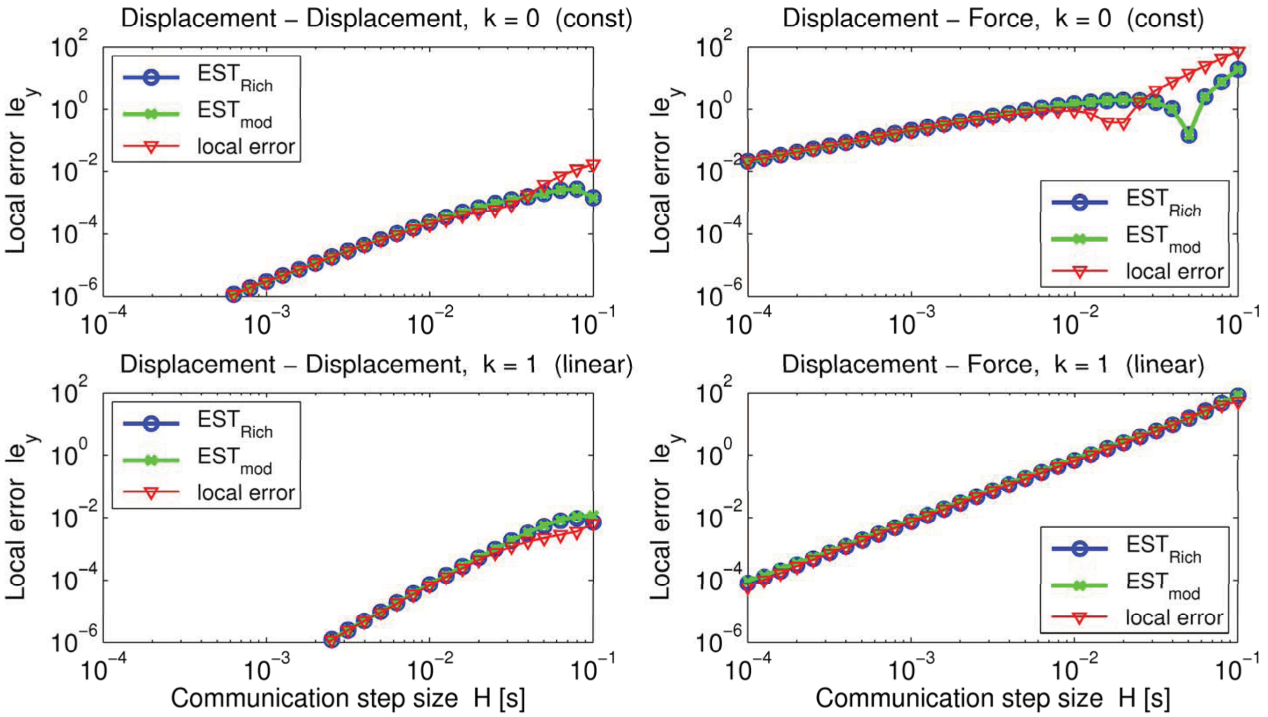
(Olsson 2011)

Modified Richardson extrapolation

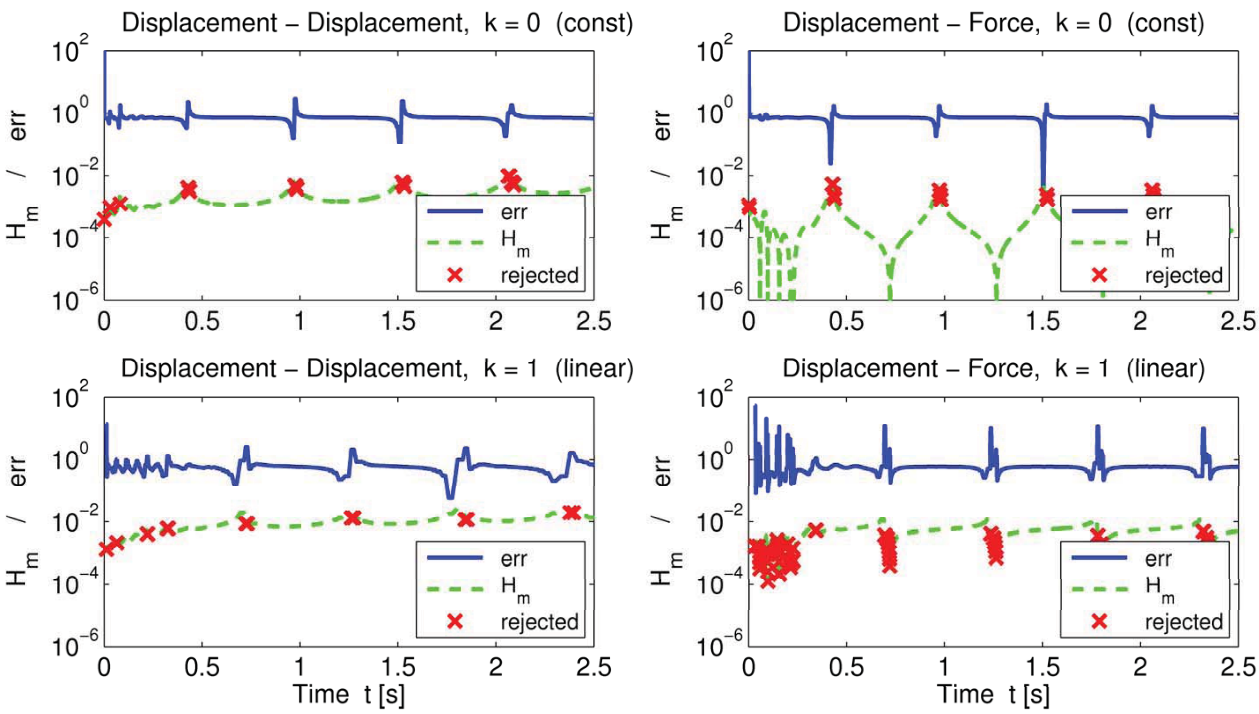
$$\text{EST}_{\text{mod}} := \frac{\tilde{\boldsymbol{y}}(T_{m+2}) - \bar{\boldsymbol{y}}(T_{m+2})}{c_{k,\text{mod}} - 1}$$

$c_{k,\text{mod}}$	without direct feed-through	with direct feed-through
$k = 0$	2.0	2.0
$k = 1$	$14/5 = 2.8$	3.0
$k = 2$	$32/9 \approx 3.6$	4.0

Local error estimates: Benchmark Quarter car



Communication step size control: Quarter car



4. Stabilization of modular time integration

Staggered time grids Park (2000), ...

- Tailored to two coupled subsystems (fluid-structure interaction)
- Communication step $T_m \to T_{m+1} = T_m + H$ for Subsystem 1 is based on (known) data at $T_{m+1/2} = T_m + H/2$ from Subsystem 2
- Macro step $T_{m+1/2} \to T_{m+3/2} = T_{m+1/2} + H$ for Subsystem 2 is based on (known) data at $T_{m+1} = T_m + H$ from Subsystem 1

Projection steps Kübler, Schiehlen (2000), [Tseng, Hulbert (1999)], ...

$$\left\{ \begin{array}{rcl} \dot{y}_i & = & \varphi_i(y_1,\ldots,y_r,w)\,, \quad (i=1,\ldots,r) \\ 0 & = & \gamma(y_1,\ldots,y_r,w) \end{array} \right.$$

Solve at $t = T_{m+1}$ the nonlinear equations $\gamma(y_{m+1,1},\ldots,y_{m+1,r},w_{m+1}) = 0$ to get $w_{m+1} \approx w(T_{m+1})$ (Broyden's method).

Overlapping modular time integration Schierz, A. (2012), ...

$$M_1\,\ddot{q}_1 = f_1(q_1,\dot{q}_1) - G_1^\top(q_1,q_2)\lambda \qquad M_2\,\ddot{q}_2 = f_2(q_2,\dot{q}_2) - G_2^\top(q_1,q_2)\lambda$$

$$0 = g(q_1(t),q_2(t)) \quad \overset{d^2/dt^2}{\Rightarrow} \quad 0 = G_1\,\ddot{q}_1 + G_2\,\ddot{q}_2 + g_{qq}(q_1,\dot{q}_1,q_2,\dot{q}_2)$$

Macro step $T_n \to T_{n+1}$: Extrapolation $\Rightarrow$ $a_1^0 \approx \ddot{q}_1$, $a_2^0 \approx \ddot{q}_2$, $\lambda^0 \approx \lambda$

$$\begin{array}{rcl} M_1 A_1 & = & f_1 - G_1^\top((I - P_1P_1^\top)\lambda^0 + P_1P_1^\top\,\Lambda_1) \\ 0 & = & P_1^\top\big(G_1 A_1 + G_2\,a_2^0 + g_{qq}(\ldots)\big) \end{array}$$

$$\begin{array}{rcl} M_2 A_2 & = & f_2 - G_2^\top((I - P_2P_2^\top)\lambda^0 + P_2P_2^\top\,\Lambda_2) \\ 0 & = & P_2^\top\big(G_1\,a_1^0 + G_2 A_2 + g_{qq}(\ldots)\big) \end{array}$$

$$\begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} a_1^0 \\ a_2^0 \\ \lambda^0 \end{pmatrix} + R(t) \begin{pmatrix} A_1 - a_1^0 \\ A_2 - a_2^0 \\ P_1P_1^\top(\Lambda_1 - \lambda^0) \\ P_2P_2^\top(\Lambda_2 - \lambda^0) \end{pmatrix}$$

Stable modular integration ($\alpha = 0$) for suitable weights $R(t)$

Overlapping modular time integration (II)

$$\begin{cases} \dot{y}_i = \varphi_i(y_1,\ldots,y_r,\underline{w}), & (i=1,\ldots,r) \\ 0 = \underline{\gamma(y_1,\ldots,y_r,w)} \end{cases}$$

Communication step

- Integrate r subsystems separately with stage functions $\underline{Y_i}$, $\underline{W_i}$.
- Assign each constraint to $l \geq 1$ subsystems $\Rightarrow$ $\underline{0 = P_i^\top \gamma(y,w)}$.

$$\left.\begin{array}{rcl} \underline{\dot{y}_i} & = & \varphi_i(\underline{Y_i},\underline{W_i}) \\ 0 & = & \underline{P_i^\top \gamma(\underline{Y_i},\underline{W_i})} \end{array}\right\},\quad (i=1,\ldots,r) \quad \Rightarrow \quad \underline{y_i},\ \underline{P_i^\top W_i}$$

- Linear combination with weights A_i :

$$w(t) = \Big(I - \sum_{i=1}^{r} A_i P_i P_i^\top \Big) w(t) + \sum_{i=1}^{r} A_i P_i P_i^\top \underline{W_i(t)}$$

Theorem There are weights $A_i(t)$ such that the contractivity condition is satisfied with $\boxed{\alpha = 0}$.

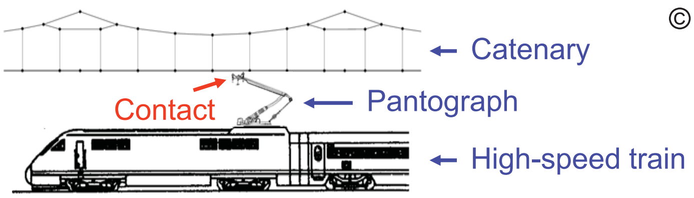
Stability of modular time integration: Example

Gauss-Seidel (standard) $\boxed{\alpha = \max \| (G_2 M_2^{-1} G_2^\top)^{-1} (G_1 M_1^{-1} G_1^\top) \| < 1}$

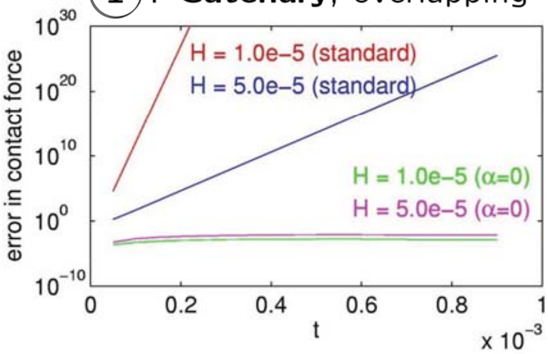
Benchmark

(A., Simeon 1998)

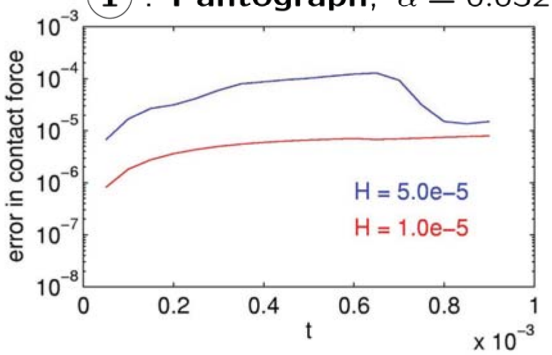
© A. Veitl (1999)



(1) : **Catenary**, overlapping



(1) : **Pantograph**, $\alpha = 0.032$



Martin Arnold (Martin Luther University Halle-Wittenberg, Germany)
Modular time integration. – Seminar at Lund University, Sweden, June 2012.

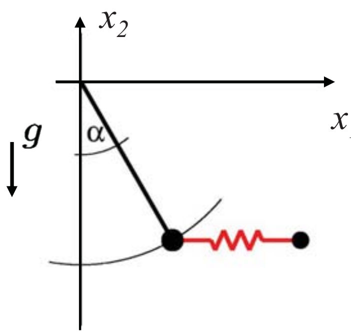
Stabilized extrapolation

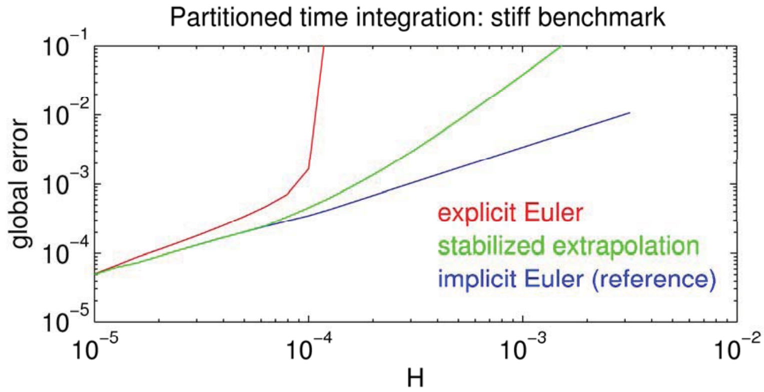
Micro step $t_n^{(m)} \to t_{n+1}^{(m)}$:

$$x_{n+1,1}^{(m)} = x_{n,1}^{(m)} + h f_1(x_{n+1,1}^{(m)}, \tilde{x}_{n+1,2}^{(m)})$$

$$x_{n+1,2}^{(m)} = x_{n,2}^{(m)} + h f_2(x_{n+1,1}^{(m)}, x_{n+1,2}^{(m)})$$

$$\tilde{x}_{n+1,2}^{(m)} := x_2^{(m)} + h\Big(I - h\frac{\partial f_2}{\partial x_2}\Big)^{-1}\Big(f_2(x_1^{(m)}, x_2^{(m)}) + \frac{\partial f_2}{\partial x_1} \cdot (x_{n+1,1}^{(m)} - x_1^{(m)})\Big)$$





Summary

Functional Mock-up Interface for Model Exchange and Co-Simulation

- Interface standard for industrial simulation tools (code export, co-simulation)
- Co-Simulation: Block-oriented master-slave framework
- Data exchange restricted to communication points: Modular time integration
- Variable communication steps, Solver dump functionality (`fmiSetFMUState`)
- Advanced interface: Higher order derivatives of block inputs, System matrices

Advanced master algorithms for co-simulation

- Higher order signal extrapolation
- Communication step size control for coupled systems without algebraic loops based on reliable local error estimates
- Numerically stable time integration of coupled stiff ODEs / DAEs using Jacobians

Acknowledgements H. Olsson (Dassault Systèmes), BMBF (01IS08002N)