BIT
2001, Vol. 41, No. 1, pp. 001–025

0006-3835/01/4101-0001 $16.00
© Swets & Zeitlinger

PRECONDITIONED DYNAMIC ITERATION FOR COUPLED DIFFERENTIAL-ALGEBRAIC SYSTEMS*

MARTIN ARNOLD[1] and MICHAEL GÜNTHER[2]

[1] *Vehicle System Dynamics Group, Institute of Aeroelasticity, DLR German Aerospace Center, P.O. Box 1116, D-82230 Wessling, Germany. email: martin.arnold@dlr.de*

[2] *Institute of Scientific Computing and Mathematical Modelling, University of Karlsruhe, Engesserstr. 6, D-76128 Karlsruhe, Germany.*
email: guenther@iwrmm.math.uni-karlsruhe.de

Abstract.

The network approach to the modelling of complex technical systems results frequently in a set of differential-algebraic systems that are connected by coupling conditions. A common approach to the numerical solution of such coupled problems is based on the coupling of standard time integration methods for the subsystems. As a unified framework for the convergence analysis of such multi-rate, multi-method or dynamic iteration approaches we study in the present paper the convergence of a dynamic iteration method with a (small) finite number of iteration steps in each window. Preconditioning is used to guarantee stability of the coupled numerical methods. The theoretical results are applied to quasilinear problems from electrical circuit simulation and to index-3 systems arising in multibody dynamics.

AMS subject classification: 65L80, 65L20, 65Y05.

Key words: Differential-algebraic systems, coupled problems, dynamic iteration, waveform relaxation.

1 Introduction.

The analysis of coupled physical phenomena leads often to coupled systems of differential equations that have to be solved numerically [17]. In the present paper we focus on the coupling of $r \geq 2$ differential-algebraic systems

$$\left.\begin{aligned} \dot{y}_i(t) &= f_i(y, z_i)\,, \\ 0 &= h_i(y, z_i, u) \end{aligned}\right\} \; (i = 1, \ldots, r) \tag{1.1a}$$

by n_u algebraic equations

$$0 = g(y, z)\,. \tag{1.1b}$$

*Received October 1999. Revised May 2000. Communicated by Christian Lubich.

The i-th subsystem of (1.1a) consists of n_{y_i} differential equations $\dot{y}_i = f_i$ and n_{z_i} algebraic equations $0 = h_i$. We assume that the initial value problem

$$y(0) = y_0\,, \qquad z(0) = z_0\,, \qquad u(0) = u_0 \tag{1.2}$$

for (1.1) has a unique solution

$$y = (y_1, \dots, y_r)^T : \; [0, T_e] \to \mathbb{R}^{n_y}, \qquad z = (z_1, \dots, z_r)^T : \; [0, T_e] \to \mathbb{R}^{n_z},$$

$u : [0, T_e] \to \mathbb{R}^{n_u}$ on the finite time interval $[0, T_e]$, $n_y = \sum n_{y_i}$, $n_z = \sum n_{z_i}$. In a neighbourhood of this solution the functions $f = (f_1, \dots, f_r)^T$, $h = (h_1, \dots, h_r)^T$, and g are supposed to be sufficiently often differentiable. Furthermore, it is supposed that the Jacobians

$$\begin{bmatrix} \frac{\partial h}{\partial z} & \frac{\partial h}{\partial u} \\ \frac{\partial g}{\partial z} & 0 \end{bmatrix}, \qquad \frac{\partial h_i}{\partial z_i}, \quad (i = 1, \dots, r) \tag{1.3}$$

are non-singular such that the coupled system (1.1) has index 1 and the equations $h_i(y, z_i, u) = 0$ are (locally) uniquely solvable w.r.t. z_i [3].

Differential-algebraic systems of the form (1.1a) result typically from a network approach if r components of a complex physical or technical system are modelled separately [13]. Typical coupling conditions are, e.g., balance equations, continuity conditions, and contact conditions (see also Section 4 below).

Instead of solving the overall system by any standard time integration method [3, 12] the special structure of this coupled system may be exploited in the numerical solution by coupling r (possibly different) numerical methods for the separate solution of the r subsystems in (1.1a). This modular approach is called *distributed time integration* of (1.1) since the time integration of the overall system is distributed to r separate time integration methods for the subsystems. For each individual subsystem a tailored time integration method may be used and stepsize and order may be adapted to the solution behaviour of this subsystem.

Furthermore, the distributed numerical solution offers a great potential for parallelism since all subsystems may be integrated in parallel on different hardware platforms, e.g., in a cluster of workstations. From the point of view of software engineering, such coupled numerical methods arise quite naturally whenever specialized simulation tools are coupled in the analysis of complex technical systems ("simulator coupling" or "co-simulation"; see, e.g., [1, 15, 23]).

To solve an initial value problem on a finite time interval $[0, T_e]$, the r subsystems are solved separately on *windows* (or *macro-steps*) $[T_n, T_{n+1}]$ with synchronization points T_n $(0 = T_0 < T_1 < \cdots < T_N = T_e)$. The integration of the overall system starts at $T_0 = 0$ and continues stepwise from one window to the next one until $T_N = T_e$ is reached. Classical approaches to the distributed time integration of ordinary differential equations (ODEs) include the use of different stepsizes in the subsystems (multi-rate methods [7, 10]), the coupling of different integration methods (multi-method approach [19]) and the iterative refinement by waveform relaxation or dynamic iteration methods [16, 18].

In the application to coupled differential-algebraic systems these techniques suffer from instability unless a contractivity condition is satisfied. This condition was introduced in 1982 by Lelarasmee et al. [16] and has been used also by Jackiewicz and Kwapisz [14] to prove convergence of dynamic iteration methods in a single window $[T_n, T_{n+1}]$. Recently, Kübler and Schiehlen [15] proposed a Newton-like method to fix the instability of the standard multi-method approach for coupled differential-algebraic systems.

In the present paper we analyse the convergence of distributed time integration methods for coupled differential-algebraic systems (1.1) in detail. We consider dynamic iteration with a *finite* number of iteration steps in each window and study the error propagation in the stepwise integration from $t = 0$ to $t = T_e$. Non-iterative techniques like multi-rate and multi-method approaches are covered by this analysis as well, since they may be interpreted as methods with one single iteration step per window.

A basic result is the observation that independent of the size $H_n := T_{n+1} - T_n$ of the windows and independent of the number k_n of iteration steps per window a contractivity condition is necessary for stability and convergence. In the stable case error estimates are given that depend strongly on H_n and k_n. If on the other hand the contractivity condition is violated then preconditioning may be used to enforce convergence. All these results are not restricted to semi-explicit index-1 systems (1.1) but apply also to more general coupled differential-algebraic systems as long as they can be transformed analytically to the form (1.1), e.g., by index reduction.

The paper is organized as follows. In Section 2 we introduce the dynamic iteration method and study its convergence and the stabilization by preconditioning. The technical details of the convergence proof are shifted to Section 3. In Section 4 the results are extended to quasilinear and to higher index systems arising in electrical circuit simulation and in multibody dynamics. Test results for a benchmark problem from mechanical engineering are given in Section 5.

2 Dynamic iteration for coupled differential-algebraic systems.

To study the basic error propagation mechanisms of distributed time integration methods we abstract in this section from the algorithmic details of multi-rate and multi-method approaches and consider instead dynamic iteration methods, i.e., iteration methods in function spaces. In Section 2.1 the considered class of coupled problems is specified in more detail. The dynamic iteration method is formulated in Section 2.2 and the error estimates are given in Section 2.3. In Section 2.4 we propose a preconditioned method for coupled differential-algebraic systems to fix the above mentioned instability phenomenon in the distributed time integration.

2.1 Problem class.

Methods for coupled systems (1.1) have to be tailored to the topology of the system that is determined by the specific structure of the coupling conditions (1.1b). Subsystems of (1.1) may be coupled by the argument y in the right hand sides of (1.1a) and/or by coupling conditions $g_j = 0$ in (1.1b). To keep the notation compact we restrict ourselves in the present paper to the coupling of $r = 2$ subsystems. However, the concepts of error analysis and preconditioning presented in Section 2.3 and Section 2.4 may be extended straightforwardly to coupled systems with a fairly general structure.

Similar to (1.3) we assume that the Jacobians

$$\left[\begin{array}{cc} \frac{\partial h_1}{\partial z_1} & \frac{\partial h_1}{\partial u} \\ \frac{\partial g}{\partial z_1} & 0 \end{array}\right] , \qquad \left[\begin{array}{cc} \frac{\partial h_2}{\partial z_2} & \frac{\partial h_2}{\partial u} \\ \frac{\partial g}{\partial z_2} & 0 \end{array}\right] \tag{2.1}$$

are non-singular in a neighbourhood of the analytical solution of (1.1). These conditions guarantee that for $i = 1$ and for $i = 2$ the systems of nonlinear equations

$$\begin{aligned} 0 &= h_i(y_1, y_2, z_i, u) \, , \\ 0 &= g(y_1, y_2, z_1, z_2) \end{aligned}$$

are (locally) uniquely solvable w. r. t. z_i and u.

2.2 *Definition of the method.*

Since our main interest is in the *coupling* of methods in the time integration of (1.1) we suppose in the following that initial value problems for the *subsystems* (1.1a) may be solved efficiently by standard methods [3, 12]. Neglecting the details of time integration for the subsystems we end up with a *numerical* solution

$$(\, \tilde{y}(t), \tilde{z}(t), \tilde{u}(t)\,)^T \; : \; [0, T_e] \to \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u}$$

for the coupled system (1.1) that is composed of *analytical* solutions of initial value problems for the subsystems. The error propagation from one window to the next one is therefore studied in function spaces.

The dynamic iteration method is formulated for coupled systems (1.1) with $r = 2$ and initial values (1.2). We consider a window $[T_n, T_{n+1}] \subset [0, T_e]$ and suppose that the numerical solution has already been computed for $t \in [0, T_n]$. In $[T_n, T_{n+1}]$ a finite number k_n of dynamic iteration steps is applied to get the numerical solution

$$\begin{gathered} \tilde{y}|_{(T_n,T_{n+1}]} := \tilde{y}_n^{(k_n)}|_{(T_n,T_{n+1}]} \, , \qquad \tilde{z}|_{(T_n,T_{n+1}]} := \tilde{z}_n^{(k_n)}|_{(T_n,T_{n+1}]} \, , \\ \tilde{u}|_{(T_n,T_{n+1}]} := \tilde{u}_n^{(k_n)}|_{(T_n,T_{n+1}]} \, . \end{gathered} \tag{2.2}$$

The iterates $\tilde{y}_n^{(k)} = (\tilde{y}_{1,n}^{(k)}, \tilde{y}_{2,n}^{(k)})^T$, $\tilde{z}_n^{(k)} = (\tilde{z}_{1,n}^{(k)}, \tilde{z}_{2,n}^{(k)})^T$, and $\tilde{u}_n^{(k)}$ with $k \geq 1$ are defined recursively by a Gauss–Seidel method that is adapted to the specific structure of (1.1). Each iteration step starts with the solution of an initial value problem for the first subsystem to get $(\, \tilde{y}_{1,n}^{(k)}, \tilde{z}_{1,n}^{(k)}\,)^T \; : \; [T_n, T_{n+1}] \to \mathbb{R}^{n_{y_1}} \times \mathbb{R}^{n_{z_1}}$:

$$\begin{aligned} \dot{\tilde{y}}_{1,n}^{(k)}(t) &= f_1(\tilde{y}_{1,n}^{(k)}, \tilde{y}_{2,n}^{(k-1)}, \tilde{z}_{1,n}^{(k)}) \, , \qquad \tilde{y}_{1,n}^{(k)}(T_n) = \tilde{y}_1(T_n) \, , \\ 0 &= h_1(\tilde{y}_{1,n}^{(k)}, \tilde{y}_{2,n}^{(k-1)}, \tilde{z}_{1,n}^{(k)}, \tilde{u}_n^{(k-1)}) \, . \end{aligned} \tag{2.3a}$$

Afterwards the iterates $(\,\tilde{y}_{2,n}^{(k)}, \tilde{z}_{2,n}^{(k)}, \tilde{u}_n^{(k)}\,)^T : [T_n, T_{n+1}] \to \mathbb{R}^{n_{y_2}} \times \mathbb{R}^{n_{z_2}} \times \mathbb{R}^{n_u}$ are obtained as solution of an initial value problem for the second subsystem:

$$\begin{aligned} \dot{\tilde{y}}_{2,n}^{(k)}(t) &= f_2(\tilde{y}_{1,n}^{(k)}, \tilde{y}_{2,n}^{(k)}, \tilde{z}_{2,n}^{(k)})\,, \qquad \tilde{y}_{2,n}^{(k)}(T_n) = \tilde{y}_2(T_n)\,, \\ 0 &= h_2(\tilde{y}_{1,n}^{(k)}, \tilde{y}_{2,n}^{(k)}, \tilde{z}_{2,n}^{(k)}, \tilde{u}_n^{(k)})\,, \\ 0 &= g(\tilde{y}_{1,n}^{(k)}, \tilde{y}_{2,n}^{(k)}, \tilde{z}_{1,n}^{(k)}, \tilde{z}_{2,n}^{(k)})\,. \end{aligned} \tag{2.3b}$$

The iteration starts with initial guesses $(\tilde{y}_n^{(0)}, \tilde{u}_n^{(0)})^T$ that may be computed extrapolating $\tilde{y}$ and $\tilde{u}$ from $(T_{n-1}, T_n]$ to $[T_n, T_{n+1}]$, (set $(T_{-1}, T_0] := \{T_0\}$ for a unified notation).

Remark 2.1.

(i) The initial values $\tilde{y}_n^{(k)}(T_n) = \tilde{y}(T_n)$ in (2.3) guarantee that (2.2) results in a continuous numerical solution $\tilde{y}$. However, the algebraic components $\tilde{z}$, $\tilde{u}$ are in general only piecewise continuous and may have discontinuities at $t = T_n$.

(ii) Equations (2.3a) form an index-1 system since $(\partial h_1)/(\partial z_1)$ is non-singular (see (1.3)). Therefore the initial value problem (2.3a) is uniquely solvable if $\tilde{y}_1(T_n)$ is sufficiently close to $y_1(T_n)$. The initial value $\tilde{z}_{1,n}^{(k)}(T_n)$ is determined by $h_1 = 0$. In the same way the unique solvability of (2.3b) is guaranteed by the non-singularity of the second block matrix in (2.1).

(iii) The Gauss–Seidel type method (2.3) is constructed such that $\tilde{z}_n^{(k)}$ does not enter the $(k+1)$-st step of iteration. This point is essential since in this way it may be guaranteed that only the errors in components y and u are propagated during the dynamic iteration and during the integration from $t = 0$ to $t = T_e$. On the other hand it is not essential to apply this Gauss–Seidel method also to the differential components y in the right hand sides of (2.3). Alternatives are Jacobi, Picard or other iteration schemes [18] that may be written in the form (2.3) substituting in the right hand sides

$$\begin{aligned} (\,\tilde{y}_{1,n}^{(k)}, \tilde{y}_{2,n}^{(k-1)}\,)^T &\to (\,(I - B_{11})\tilde{y}_{1,n}^{(k)} + B_{11}\tilde{y}_{1,n}^{(k-1)},\ \tilde{y}_{2,n}^{(k-1)}\,)^T, \\ (\,\tilde{y}_{1,n}^{(k)}, \tilde{y}_{2,n}^{(k)}\,)^T &\to (\,(I - B_{21})\tilde{y}_{1,n}^{(k)} + B_{21}\tilde{y}_{1,n}^{(k-1)},\ (I - B_{22})\tilde{y}_{2,n}^{(k)} + B_{22}\tilde{y}_{2,n}^{(k-1)}\,)^T. \end{aligned} \tag{2.4}$$

The splitting matrices are, e.g., $B_{11} = B_{21} = 0$, $B_{22} = 0$ for Gauss–Seidel iteration and $B_{11} = 0$, $B_{21} = I$, $B_{22} = 0$ for Jacobi iteration. All results that will be derived in the following sections for (2.3) remain valid in the more general setting (2.4).

In the convergence analysis a more formal notation of the dynamic iteration method will be helpful. Since $\tilde{z}_n^{(k)}$ is used only locally inside the k-th step of iteration the dynamic iteration (2.3) defines a mapping

$$\begin{bmatrix} \tilde{y}_n^{(k-1)} \\ \tilde{u}_n^{(k-1)} \end{bmatrix} \mapsto \begin{bmatrix} \tilde{y}_n^{(k)} \\ \tilde{u}_n^{(k)} \end{bmatrix} =: \Psi_n\left(\begin{bmatrix} \tilde{y}_n^{(k-1)} \\ \tilde{u}_n^{(k-1)} \end{bmatrix} \right) \quad \text{with} \quad \Psi_n = \begin{bmatrix} \Psi_{y,n} \\ \Psi_{u,n} \end{bmatrix}, \tag{2.5}$$

$\Psi_n : C_n^{1,0} \to C_n^{1,0}$ and $C_n^{1,0} := C^1([T_n, T_{n+1}], \mathbb{R}^{n_y}) \times C([T_n, T_{n+1}], \mathbb{R}^{n_u})$. In this more general setting the initial values $\tilde{y}_n^{(k)}(T_n)$ are given by $\tilde{y}_n^{(k-1)}(T_n)$, i.e., by the first argument of Ψ_n.

The iteration starts with

$$\begin{bmatrix} \tilde{y}_n^{(0)} \\ \tilde{u}_n^{(0)} \end{bmatrix} := \Phi_n \left(\begin{bmatrix} \tilde{y}|_{(T_{n-1},T_n]} \\ \tilde{u}|_{(T_{n-1},T_n]} \end{bmatrix} \right) = \begin{bmatrix} \Phi_{y,n}(\tilde{y}|_{(T_{n-1},T_n]}) \\ \Phi_{u,n}(\tilde{u}|_{(T_{n-1},T_n]}) \end{bmatrix} \tag{2.6}$$

where $\Phi_n : \bar{C}_{n-1}^{1,0} \to C_n^{1,0}$ denotes an operator that extrapolates $(\tilde{y}, \tilde{u})$ continuously from $(T_{n-1}, T_n]$ to $[T_n, T_{n+1}]$, $\bar{C}_n^{1,0} := \{\, (y, u)|_{(T_n,T_{n+1}]} \,:\, (y, u) \in C_n^{1,0} \,\}$. With these notations (2.2) may be written as

$$\begin{bmatrix} \tilde{y}|_{(T_n,T_{n+1}]} \\ \tilde{u}|_{(T_n,T_{n+1}]} \end{bmatrix} = (\Psi_n^{k_n} \circ \Phi_n) \left(\begin{bmatrix} \tilde{y}|_{(T_{n-1},T_n]} \\ \tilde{u}|_{(T_{n-1},T_n]} \end{bmatrix} \right) \Bigg|_{(T_n,T_{n+1}]} . \tag{2.7}$$

Error estimates will be given in the L^∞–norm $\|\cdot\|_{[T_m,T_n]}$ with $[T_m, T_n]$ denoting the time interval under consideration. If this time interval is obvious from the context, then the index "$[T_m, T_n]$" is omitted.

2.3 Error estimates.

In the error analysis of (2.2)–(2.3) we have to study the error of the iterates in each single window $[T_n, T_{n+1}]$ and the error propagation from one window to the next one. To simplify notation we assume in the following that $H_n = H = \text{const}$ throughout integration. All results remain valid for variable window sizes H_n as long as their ratios for subsequent windows satisfy $\underline{H} \le H_{n+1}/H_n \le \overline{H}$ with constants $\underline{H}$ and $\overline{H}$.

The accuracy of the initial guesses $(\tilde{y}_n^{(0)}, \tilde{u}_n^{(0)})^T$ is defined by Φ_n:

Definition 2.1. *The error of the extrapolated analytical solution of* (1.1) *defines the* extrapolation errors

$$\delta y_n := \Phi_{y,n}(y|_{(T_{n-1},T_n]}) - y|_{[T_n,T_{n+1}]} , \qquad \delta u_n := \Phi_{u,n}(u|_{(T_{n-1},T_n]}) - u|_{[T_n,T_{n+1}]}$$

of the distributed time integration method.

The most simple initial guesses are constant functions $\tilde{y}_n^{(0)}(t) = \tilde{y}(T_n)$, $\tilde{u}_n^{(0)}(t) = \tilde{u}(T_n)$ resulting in $\|\delta y_n\| = \mathcal{O}(H)$, $\|\delta u_n\| = \mathcal{O}(H)$. Approximations of higher order may be obtained using higher degree polynomials.

Lemma 2.1. *Let $\pi(x; t_0, \ldots, t_m)$ denote the polynomial interpolating $x(t)$ at $t_0, \ldots, t_m$ ($\deg \pi \le m$). If $n \ge 1$ and polynomials with $m+1$ pairwise distinct interpolation points $t_0, \ldots, t_m \in (T_{n-1}, T_n]$ and $t_m = T_n$ are used as initial guesses in $[T_n, T_{n+1}]$ then Φ_n is given by*

$$\Phi_{y,n}(Y) = \pi(Y; t_0, \ldots, t_m) , \quad \Phi_{u,n}(U) = \pi(U; t_0, \ldots, t_m)$$

and we get:

(i) *$\|\delta y_n\| = \mathcal{O}(H^{m+1})$, $\|\delta u_n\| = \mathcal{O}(H^{m+1})$.*

(ii) *Φ_n extrapolates continuously from $(T_{n-1}, T_n]$ to $[T_n, T_{n+1}]$.*

(iii) *$\Phi_{y,n}$ and $\Phi_{u,n}$ satisfy uniform Lipschitz conditions (i.e., the Lipschitz constants may depend on m, t_0, …, t_m but they are independent of H).*

Proof. (i) and (ii) are trivial and (iii) may be seen writing π in Lagrangian form with basis polynomials $L_j^{(m)}(t;t_0,\dots,t_m)$:

$$\begin{aligned}
&\|\pi[\tilde{x};t_0,\dots,t_m]-\pi[x;t_0,\dots,t_m]\|_{[T_n,T_{n+1}]}\\
&\quad=\Big\|\sum_{j=0}^{m}L_j^{(m)}(t;t_0,\dots,t_m)\cdot(\tilde{x}(t_j)-x(t_j))\Big\|_{[T_n,T_{n+1}]}\\
&\quad\le\sum_{j=0}^{m}\max_{T_n\le t\le T_{n+1}}\big|L_j^{(m)}(t;t_0,\dots,t_m)\big|\cdot\|\tilde{x}-x\|_{(T_{n-1},T_n]}\le L\|\tilde{x}-x\|_{(T_{n-1},T_n]}
\end{aligned}$$

with a constant L that depends on m, t_0, …, t_m but is independent of H. (In the case of variable window sizes the constant L depends additionally on the bounds $\underline{H}$, $\overline{H}$ for the stepsize ratios H_{n+1}/H_n.) □

Applying Gauss–Seidel, Jacobi or Picard like dynamic iteration methods to coupled ODEs convergence may always be achieved using sufficiently small window sizes H [18]. In the application to coupled differential-algebraic systems (1.1), however, the additional *contractivity condition* $\alpha<1$ has to be satisfied to guarantee a stable error propagation in the algebraic components u. Here, α is defined by

$$\alpha:=\max_{t\in[0,T]}\|R_2^{-1}(t)R_1(t)\| \tag{2.8}$$

with

$$R_i(t):=\Big[\frac{\partial g}{\partial z_i}\Big(\frac{\partial h_i}{\partial z_i}\Big)^{-1}\frac{\partial h_i}{\partial u}\Big](y(t),z(t),u(t))\,,\quad(\,i=1,2\,)\,. \tag{2.9}$$

Theorem 2.2. *Consider the dynamic iteration method* (2.3) *with extrapolation operators* Φ_n *that extrapolate continuously from* $(T_{n-1},T_n]$ *to* $[T_n,T_{n+1}]$. *If*

(i) $\|\delta y_n\|=\mathcal{O}(H)$, $\|\delta u_n\|=\mathcal{O}(H)$,

(ii) $\Phi_{y,n}$ *and* $\Phi_{u,n}$ *satisfy uniform Lipschitz conditions with constant* L, *and*

(iii) $\alpha\le\bar{\alpha}$ *and* $L\alpha^{k_n}\le\bar{\alpha}$ *for all* $n\ge0$ *with* $nH\le T_N=T_e$ *and a constant* $\bar{\alpha}<1$

then there are constants $C^*,H_0>0$ *and* $\mu=\alpha+\mathcal{O}(H)$ *such that for all* $H\le H_0$

$$\begin{aligned}
&\|\tilde{y}-y\|_{[0,T_e]}+\|\tilde{z}-z\|_{[0,T_e]}+\|\tilde{u}-u\|_{[0,T_e]}\\
&\qquad\le C^*\cdot\max_{0\le n<N}\big(\mu^{\max(0,k_n-2)}\|\delta y_n\|+\mu^{k_n-1}\|\delta u_n\|\big)\,.
\end{aligned}\tag{2.10}$$

Proof. See Section 3. □

Remark 2.2.

(i) The contractivity condition $\alpha\le\bar{\alpha}<1$ guarantees the convergence of the dynamic iteration for $k\to\infty$ in a single window $[T_n,T_{n+1}]$ [14, 16].

(ii) The numbering of the subsystems in (1.1) is significant since it determines their order in the Gauss–Seidel method (2.3) and therefore also the contractivity constant α; see (2.8). The subsystems should be arranged such that α gets as small as possible.

(iii) Estimate (2.10) shows that in the limit case $H \to 0$ a very small global error may be obtained using high order extrapolation operators Φ_n. However, in practical computations low order extrapolation (order ≤ 2) is often more favourable since high order extrapolation results typically in large constants in the $\mathcal{O}(\cdot)$-terms in $\|\delta y_n\|, \|\delta u_n\| = \mathcal{O}(H^{m+1})$ and in a large Lipschitz constant L; see Lemma 2.1.
(iv) The contractivity condition $L\alpha^{k_n} \leq \bar{\alpha} < 1$ indicates that the stability of the dynamic iteration method may be strongly influenced by the Lipschitz constant L of the extrapolation operators Φ_n (see Example 2.1 below).

Example 2.1. Consider (1.1) with $r = 2$,

$$f_1 = 1\,,\quad f_2 = 0\,,\quad h_1 = (\alpha - 1)y_1 + \alpha z_1 - \alpha u\,,\quad h_2 = \alpha z_2 - u\,,\quad g = z_1 - z_2$$

and initial values $y(0) = 0$. The analytical solution is $y_1 = t$, $y_2 = 0$, $z_1 = t/\alpha$, $z_2 = t/\alpha$, $u = t$. The parameter $\alpha \in (0, 1)$ is identical with the contractivity constant in (2.8). This coupled system is integrated by (2.3) with $H_n = H = \text{const}$, $k_n = \bar{k} = \text{const}$ and initial guesses $\tilde{y}_n^{(0)}(t) = \tilde{y}(T_n)$ and

$$\tilde{u}_n^{(0)}(t) = \tilde{u}(T_n) + \beta\,\frac{\tilde{u}(T_{n-1} + cH) - \tilde{u}(T_n)}{T_{n-1} + c \cdot H - T_n}(t - T_n)\,,\quad (\, t \in [T_n, T_{n+1}]\,) \tag{2.11}$$

that depend on parameters $c \in (0, 1)$ and $\beta \in \mathbb{R}$. After some straightforward computations we get

$$v_{n+1} = \beta\alpha^{\bar{k}} v_n + (1 - \alpha^{\bar{k}})(1 - c)H \quad \text{with} \quad v_n := \tilde{u}(T_n) - \tilde{u}(T_{n-1} + cH)\,.$$

The solution v_n of this difference equation gets unstable if $|\beta|\alpha^{\bar{k}} > 1$. Therefore $\bar{k} \geq \log|\beta| / \log(1/\alpha)$ iteration steps have to be performed in each window to get a stable error propagation from $t = 0$ to $t = T_e$. (Note, however, that for the most obvious choices $\beta = 0$ and $\beta = 1$ in (2.11) convergence is always guaranteed if $\alpha < 1$.)

2.4 Preconditioning.

The convergence analysis for the dynamic iteration method (2.3) shows that a contractivity constant $\alpha > 1$ may result in instability (see also Section 3 below). Preconditioning offers a convenient way to avoid this instability while keeping the simple basic structure of (2.3).

Definition 2.2. *A* preconditioner *for the dynamic iteration method* (2.3) *is a matrix function* $A : [0, T_e] \to \mathbb{R}^{n_u \times n_u}$ *such that* $I_{n_u} - A(t)$ *is non-singular for all* $t \in [0, T_e]$. *The* preconditioned *dynamic iteration is given by* (2.3) *with* $0 = h_2(\tilde{y}_{1,n}^{(k)}, \tilde{y}_{2,n}^{(k)}, \tilde{z}_{2,n}^{(k)}, \tilde{u}_n^{(k)})$ *being substituted by*

$$\begin{array}{rcl} 0 & = & h_2(\tilde{y}_{1,n}^{(k)}(t), \tilde{y}_{2,n}^{(k)}(t), \tilde{z}_{2,n}^{(k)}(t), U_n^{(k)}(t))\,, \\ U_n^{(k)}(t) & = & (I - A(t))\,\tilde{u}_n^{(k)}(t) + A(t)\,\tilde{u}_n^{(k-1)}(t)\,. \end{array} \tag{2.12}$$

The original method (2.3) is covered by this notation setting $A(t) \equiv 0$. If $A(t) \neq 0$ then (2.3) and (2.12) may be interpreted as (over-)relaxation method with a matrix valued relaxation parameter $A(t)$. Since the term overrelaxed dynamic iteration (*overrelaxed waveform relaxation*) might cause confusion we prefer the terminology of Definition 2.2. Note, however, that Definition 2.2 does not fit into the framework of the frequently cited work of Burrage et al. [4] who considered several approaches to preconditioning in dynamic iteration methods for ordinary differential equations.

With (2.12) and the matrix functions $R_i(t)$ of (2.9) the contractivity constant α of the preconditioned method is given by

$$\alpha := \max_{t\in[0,T]} \big\| \big(I - A(t)\big)^{-1} \big(A(t) + R_2^{-1}(t) R_1(t)\big) \big\| \tag{2.13}$$

and the preconditioner should be defined such that $\alpha < 1$. Obviously, the definition

$$A(t) := -\Big[\Big(\frac{\partial g}{\partial z_2} \Big(\frac{\partial h_2}{\partial z_2} \Big)^{-1} \frac{\partial h_2}{\partial u} \Big)^{-1} \Big(\frac{\partial g}{\partial z_1} \Big(\frac{\partial h_1}{\partial z_1} \Big)^{-1} \frac{\partial h_1}{\partial u} \Big) \Big] (y(t), z(t), u(t)) \tag{2.14}$$

results in the optimal value $\alpha = 0$. In a practical implementation of (2.14) the (unknown) arguments (y, z, u) of the Jacobians are substituted by $(\tilde{y}_n^{(k)}, \tilde{z}_n^{(k)}, \tilde{u}_n^{(k)})$.

The matrix function $A(t)$ of (2.14) may really be used as preconditioner since $I - A(t)$ is non-singular. This can be seen from the assumptions of Section 1 where we supposed that the block matrix in (1.3) is non-singular. Therefore the Schur complement of $\partial h / \partial z$ in this block matrix is non-singular, too. This implies the non-singularity of $I - A(t)$ since the Schur complement is the second factor in the factorization

$$I - A(t) = -\Big(\frac{\partial g}{\partial z_2} \Big(\frac{\partial h_2}{\partial z_2} \Big)^{-1} \frac{\partial h_2}{\partial u} \Big)^{-1} \Big(-\frac{\partial g}{\partial z_2} \Big(\frac{\partial h_2}{\partial z_2} \Big)^{-1} \frac{\partial h_2}{\partial u} - \frac{\partial g}{\partial z_1} \Big(\frac{\partial h_1}{\partial z_1} \Big)^{-1} \frac{\partial h_1}{\partial u} \Big) .$$

The convergence analysis of the preconditioned method (2.3) and (2.12) follows step by step the proof of Theorem 2.2 (Equation (3.3) in the proof of Lemma 3.1 gets slightly more complicated since $A(t)$ is involved). As a consequence of preconditioning the contractivity constant α is not longer fixed but depends now on the preconditioner $A(t)$. Therefore it is important to derive error bounds that are uniform w.r.t. α and remain valid also in the limit case $\alpha = 0$.

Corollary 2.3. *Let the assumptions of Theorem* 2.2 *be satisfied with* α *defined by* (2.13). *Then the error of the preconditioned dynamic iteration method* (2.3) *and* (2.12) *is bounded by* (2.10) *with*

$$\mu = \mu(\alpha, H) = \alpha + C_0 \, \frac{H}{\alpha + \sqrt{H}}$$

and constants C_0, C^*, *and* H_0 *that are independent of* α *but may depend on* $\bar{\alpha}$.

Proof. See Section 3. □

Remark 2.3. Corollary 2.3 shows that convergence of the preconditioned method could always be enforced using the preconditioner (2.14). Because of its rather complicated structure other preconditioners may, however, be more efficient.

3 Proof of the convergence theorem.

The solutions of initial value problems for index-1 systems depend continuously on input data. This fact may be used to prove that the iteration operators Ψ_n satisfy a Lipschitz condition. Estimates for the Lipschitz constants of Ψ_n and Ψ_n^k will be given in Lemmas 3.1 and 3.2, respectively. Finally, error propagation and error accumulation during integration are studied following standard arguments of the theory of differential-algebraic systems.

To prove Lipschitz conditions for Ψ_n and Ψ_n^k, the operators are applied to functions (Y, U) and $(\hat{Y}, \hat{U})$ close to the analytical solution (y, u). The difference of the images is denoted by

$$\begin{bmatrix} \Delta_y^{(k)} \\ \Delta_u^{(k)} \end{bmatrix} := \Psi_n^k\left(\begin{bmatrix} \hat{Y} \\ \hat{U} \end{bmatrix}\right) - \Psi_n^k\left(\begin{bmatrix} Y \\ U \end{bmatrix}\right) .$$

Lemma 3.1. *Consider the neighbourhood*

$$\mathcal{U}_\gamma := \{ (Y, U) \in C_n^{1,0} \, : \, \|Y - y|_{[T_n,T_{n+1}]}\| + \|U - u|_{[T_n,T_{n+1}]}\| \le \gamma \}$$

of the analytical solution (y, u). There are positive constants C, γ_0, and H_0 such that the estimate

$$\begin{bmatrix} \|\Delta_y^{(1)}\| \\ \|\Delta_u^{(1)}\| \end{bmatrix} \le \begin{bmatrix} CH & CH \\ C & \hat{\alpha} + CH \end{bmatrix} \cdot \begin{bmatrix} \|\hat{Y} - Y\| \\ \|\hat{U} - U\| \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \|\hat{Y}(T_n) - Y(T_n)\| \tag{3.1}$$

is satisfied for all functions $(Y, U), (\hat{Y}, \hat{U}) \in \mathcal{U}_{\gamma_0}$ and for all $H \le H_0$.
Inequality (3.1) *has to be read componentwise and contains a constant*

$$\hat{\alpha} = \alpha + \mathcal{O}(1) \left(\|\hat{Y} - y\| + \|\hat{U} - u\| + \|Y - y\| + \|U - u\| \right) \ge 0$$

with α being defined in (2.8).
(For simplicity of presentation we suppose in the following $C > \hat{\alpha}$).

Proof. Consider functions $(Y, U), (\hat{Y}, \hat{U}) \in \mathcal{U}_\gamma$ with a sufficiently small $\gamma > 0$. Then insert the functions Y_2 and U as $\tilde{y}_{2,n}^{(k-1)}$ and $\tilde{u}_n^{(k-1)}$ into (2.3) and denote the resulting functions $\tilde{y}_n^{(k)}$, $\tilde{z}_n^{(k)}$, $\tilde{u}_n^{(k)}$ by ψ_y, ψ_z, ψ_u. In the same way functions $\hat{\psi}_y$, $\hat{\psi}_z$, and $\hat{\psi}_u$ are defined inserting $\hat{Y}_2$ and $\hat{U}$. An estimate for $\Delta_y^{(1)} = \hat{\psi}_y - \psi_y$ is obtained similar to the estimates for classical Picard–Lindelöf iteration in ODE theory (see, e.g., [11, Section I.8]) since the regularity assumptions on (1.1) guarantee that f_1, f_2 satisfy Lipschitz conditions w.r.t. y and z. We get

$$\|\Delta_y^{(1)}\| = \|\hat{\psi}_y - \psi_y\| \le \|\hat{Y}(T_n) - Y(T_n)\| + \mathcal{O}(H)(\|\hat{Y} - Y\| + \|\hat{\psi}_z - \psi_z\|) . \tag{3.2}$$

For a fixed time t the algebraic equations in (2.3) are summarized to $F(0) = F(1) = 0$ with

$$F(\vartheta) := \begin{bmatrix} h_1(\hat{\psi}_{y_1}^\vartheta, \hat{Y}_2^\vartheta, \hat{\psi}_{z_1}^\vartheta, \hat{U}^\vartheta) \\ h_2(\hat{\psi}_{y_1}^\vartheta, \hat{\psi}_{y_2}^\vartheta, \hat{\psi}_{z_2}^\vartheta, \hat{\psi}_u^\vartheta) \\ g(\hat{\psi}_{y_1}^\vartheta, \hat{\psi}_{y_2}^\vartheta, \hat{\psi}_{z_1}^\vartheta, \hat{\psi}_{z_2}^\vartheta) \end{bmatrix} , \quad \vartheta \in [0, 1]$$

and $\hat{\psi}^{\vartheta}_{y_1} := (1-\vartheta)\psi_{y_1} + \vartheta\hat{\psi}_{y_1}, \ldots$. The identity $F(1) - F(0) = \int_0^1 F'(\vartheta)\,\mathrm{d}\vartheta$ gives

$$
0 = \int_0^1 \left(\begin{bmatrix} \frac{\partial h_1}{\partial z_1} & 0 & 0 \\ 0 & \frac{\partial h_2}{\partial z_2} & \frac{\partial h_2}{\partial u} \\ \frac{\partial g}{\partial z_1} & \frac{\partial g}{\partial z_2} & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\psi}_{z_1} - \psi_{z_1} \\ \hat{\psi}_{z_2} - \psi_{z_2} \\ \hat{\psi}_u - \psi_u \end{bmatrix} + \begin{bmatrix} \frac{\partial h_1}{\partial u} \\ 0 \\ 0 \end{bmatrix} \cdot (\hat{U} - U) \right) \mathrm{d}\vartheta \tag{3.3}
$$
$$
+ \mathcal{O}(1) \left(\|\hat{Y}_2 - Y_2\| + \|\hat{\psi}_y - \psi_y\| \right).
$$

In (3.3) we have omitted the arguments $\hat{\psi}^{\vartheta}_{y_1}, \hat{\psi}^{\vartheta}_{y_2}, \ldots$ of the Jacobians that are in a neighbourhood of size $\mathcal{O}(\gamma)$ of $(y(t), z(t), u(t))$. If $\gamma > 0$ is sufficiently small then (3.3) may be solved w.r.t. $\hat{\psi}_z - \psi_z$ and $\hat{\psi}_u - \psi_u$ since the matrix of coefficients has a lower diagonal 2×2 block structure with diagonal blocks that are nonsingular by assumption (see (1.3) and (2.1)). Using the expression

$$
- \begin{bmatrix} \frac{\partial h_1}{\partial z_1} & 0 & 0 \\ 0 & \frac{\partial h_2}{\partial z_2} & \frac{\partial h_2}{\partial u} \\ \frac{\partial g}{\partial z_1} & \frac{\partial g}{\partial z_2} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial h_1}{\partial u} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -(\frac{\partial h_1}{\partial z_1})^{-1} \frac{\partial h_1}{\partial u} \\ (\frac{\partial h_2}{\partial z_2})^{-1} \frac{\partial h_2}{\partial u} R_2^{-1} R_1 \\ -R_2^{-1} R_1 \end{bmatrix}
$$

with the matrix functions R_i from (2.9) we get

$$
\|\hat{\psi}_z - \psi_z\| \le \mathcal{O}(1) \left(\|\hat{Y} - Y\| + \|\hat{U} - U\| + \|\hat{\psi}_y - \psi_y\| \right), \tag{3.4}
$$

$$
\|\Delta_u^{(1)}\| = \|\hat{\psi}_u - \psi_u\| \le \hat{\alpha} \cdot \|\hat{U} - U\| + \mathcal{O}(1) \left(\|\hat{Y} - Y\| + \|\hat{\psi}_y - \psi_y\| \right). \tag{3.5}
$$

Inserting (3.4) in (3.2) estimate (3.1) is proven for components $\Delta_y^{(1)}$. The proof is completed substituting this expression in the right hand side of (3.5). □

REMARK 3.1. In the stiff case the constant C in (3.1) is very large if a classical Lipschitz condition is used to get (3.2). However, estimates with constants C of moderate size may be obtained using, e.g., one-sided Lipschitz conditions [11, Section I.10].

LEMMA 3.2. *Let the assumptions of Lemma* 3.1 *be satisfied and assume furthermore* $\hat{\alpha} < 1$ *and* $C > \hat{\alpha}$. *Then there is a constant* $\hat{C}$ *such that the estimate*

$$
\begin{bmatrix} \|\Delta_y^{(k)}\| \\ \|\Delta_u^{(k)}\| \end{bmatrix} \le \begin{bmatrix} C(4C+1)H\hat{\mu}^{\max(0,k-2)} & 4CH\hat{\mu}^{k-1} \\ 4C\hat{\mu}^{k-1} & \hat{\mu}^k + (\hat{\mu} - \hat{\alpha})^k \end{bmatrix} \cdot \begin{bmatrix} \|\hat{Y} - Y\| \\ \|\hat{U} - U\| \end{bmatrix} \tag{3.6}
$$
$$
+ \begin{bmatrix} 1 + \hat{C}H \\ \hat{C} \end{bmatrix} \cdot \|\hat{Y}(T_n) - Y(T_n)\|
$$

with

$$
\hat{\mu} = \hat{\mu}(\hat{\alpha}, H) := \hat{\alpha} + \frac{2CH}{\frac{\hat{\alpha}}{2C} + \sqrt{H}}
$$

is satisfied for all $k \ge 1$ *and for all* $H \le H_0$.

Proof. Lemma 3.1 shows that the iteration error is mainly determined by powers of matrix

$$J = \begin{bmatrix} CH & CH \\ C & \hat{\alpha} + CH \end{bmatrix} = D \begin{bmatrix} c & \sqrt{b^2 - a^2} \\ \sqrt{b^2 - a^2} & 2a + c \end{bmatrix} D^{-1} \tag{3.7}$$

with the diagonal matrix $D := \mathrm{diag}(1, 1/\sqrt{H})$ and

$$a := \hat{\alpha}/2 \,, \quad b := \sqrt{\hat{\alpha}^2/4 + C^2 H} \,, \quad c := CH \,.$$

Recursive application of (3.1) results in

$$\begin{bmatrix} \|\Delta_y^{(k)}\| \\ \|\Delta_u^{(k)}\| \end{bmatrix} \leq J^k \begin{bmatrix} \|\hat{Y} - Y\| \\ \|\hat{U} - U\| \end{bmatrix} + \sum_{i=0}^{k-1} J^i \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \|\hat{Y}(T_n) - Y(T_n)\| \,. \tag{3.8}$$

A comparison of (3.6) and (3.8) shows that bounds for the elements of J^k and of the first column of $\sum J^i$ have to be derived to prove the lemma. To analyse the powers of J the matrix is diagonalized. The two different real eigenvalues of J are given by

$$\mu_{1/2} = \frac{\hat{\alpha}}{2} + CH \pm \sqrt{\frac{\hat{\alpha}^2}{4} + C^2 H} \,, \quad \mu_1 = a + c - b \,, \;\; \mu_2 = a + c + b \,. \tag{3.9}$$

If $H > 0$ is sufficiently small then $\mu_1 < 0$ and $\hat{\mu}(\hat{\alpha}, H)$ is an upper bound for $|\mu_1|$ and $|\mu_2|$ since

$$-\frac{\hat{\alpha}}{2} + \sqrt{\frac{\hat{\alpha}^2}{4} + C^2 H} = \frac{C^2 H}{\frac{\hat{\alpha}}{2} + \sqrt{\frac{\hat{\alpha}^2}{4} + C^2 H}} \leq \frac{CH}{\frac{\hat{\alpha}}{2C} + \sqrt{H}} \,.$$

Using the factorization $J = DT\Lambda T^{-1} D^{-1}$ with $\Lambda = \mathrm{diag}(\mu_1, \mu_2)$ and an orthogonal matrix T whose columns are given by the eigenvectors of $D^{-1} J D$ we get

$$\begin{aligned} J^i &= D \, \frac{1}{2b(a+b)} \begin{bmatrix} \mu_1^i (a+b)^2 + \mu_2^i (b^2 - a^2) & (\mu_2^i - \mu_1^i)(a+b)\sqrt{b^2 - a^2} \\ (\mu_2^i - \mu_1^i)(a+b)\sqrt{b^2 - a^2} & \mu_1^i (b^2 - a^2) + \mu_2^i (a+b)^2 \end{bmatrix} D^{-1}, \\ &= D \begin{bmatrix} \frac{b^2 - a^2 - ac}{2b} (\mu_2^{i-1} - \mu_1^{i-1}) + \frac{c}{2}(\mu_1^{i-1} + \mu_2^{i-1}) & \frac{\sqrt{b^2 - a^2}}{2b} (\mu_2^i - \mu_1^i) \\ \frac{\sqrt{b^2 - a^2}}{2b} (\mu_2^i - \mu_1^i) & \frac{1}{2}(\mu_1^i + \mu_2^i) + \frac{a}{2b}(\mu_2^i - \mu_1^i) \end{bmatrix} D^{-1}. \end{aligned}$$

Restricting ourselves to sufficiently small $H > 0$ we have $|\mu_1| \leq \hat{\mu}$, $|\mu_2| \leq \hat{\mu}$, and $\hat{\mu} < 1$ such that

$$\begin{aligned} \Big| \frac{\sqrt{b^2 - a^2}}{2b} (\mu_2^i - \mu_1^i) \Big| &\leq \Big(\frac{\sqrt{b^2 - a^2}}{b} \hat{\mu} \Big) \hat{\mu}^{i-1} \\ &\leq \Bigg(C\sqrt{H} \frac{\hat{\alpha}}{\sqrt{\frac{\hat{\alpha}^2}{4} + C^2 H}} + \frac{2CH}{\frac{\hat{\alpha}}{2C} + \sqrt{H}} \Bigg) \hat{\mu}^{i-1} \\ &\leq (2C\sqrt{H} + 2C\sqrt{H}) \hat{\mu}^{i-1} \leq 4C\sqrt{H} \, \hat{\mu}^{i-1} \end{aligned}$$

and (for $i \geq 2$)

$$\begin{aligned} \left| \frac{b^2-a^2-ac}{2b} (\mu_2^{i-1} - \mu_1^{i-1}) \right| &\leq \sqrt{b^2-a^2} \cdot \left| \frac{\sqrt{b^2-a^2}}{2b} (\mu_2^{i-1} - \mu_1^{i-1}) \right| \\ &\leq 4C^2 H \hat{\mu}^{i-2} \, , \\ \left| \frac{c}{2} (\mu_1^{i-1} + \mu_2^{i-1}) \right| &\leq c\hat{\mu} \cdot \hat{\mu}^{i-2} \leq CH\hat{\mu}^{i-2} \, . \end{aligned}$$

These estimates are used as bounds for the off-diagonal elements of J^i and for the upper left element (if $i \geq 2$). For $i = 1$ this upper left element is CH. A bound for the lower right element of J^i is given by $|\mu_1|^i + |\mu_2|^i \leq (\hat{\mu} - \hat{\alpha})^i + \hat{\mu}^i$.

Therefore J^k in (3.8) is bounded elementwise by

$$\text{(3.10)} \quad \begin{bmatrix} C(4C+1)H(\hat{\mu}(\hat{\alpha}, H))^{\max(0,k-2)} & 4CH\hat{\mu}^{k-1}(\hat{\alpha}, H) \\ 4C\hat{\mu}^{k-1}(\hat{\alpha}, H) & \hat{\mu}^k(\hat{\alpha}, H) + (\hat{\mu}(\hat{\alpha}, H) - \hat{\alpha})^k \end{bmatrix} .$$

This proves the estimate (3.6) for the first term on the right hand side of (3.8). Bound (3.10) may also be used to estimate the second term in (3.8). We get

$$\text{(3.11)} \sum_{i=0}^{k-1} J^i \begin{bmatrix} 1 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 1 + CH + C(4C+1)H(1 + \hat{\mu} + \cdots + \hat{\mu}^{k-3}) \\ 4C(1 + \hat{\mu} + \cdots + \hat{\mu}^{k-3}) \end{bmatrix} \leq \begin{bmatrix} 1 + \hat{C}H \\ \hat{C} \end{bmatrix}$$

with $\hat{C} := 4C(C+1)/(1-\hat{\mu})$. Inserting (3.10) and (3.11) in (3.8) the proof is completed. □

Remark 3.2.

(i) The analytical solution (y, u) is a fixed point of Ψ_n. Therefore estimate (3.6) shows that the sequence of iterates remains in a small neighbourhood of (y, u) if $H > 0$ is sufficiently small and the assumptions of Lemma 3.2 are satisfied.

(ii) Since α depends on the preconditioner (see (2.13)) the error estimates in Lemmas 3.1 and 3.2 have been formulated uniformly w.r.t. α with constants C, $\hat{C}$, and H_0 being independent of α. For fixed $\alpha > 0$ we have $\hat{\mu}(\alpha, H) = \alpha + \mathcal{O}(H)$. However, the constant in the $\mathcal{O}(\cdot)$–term is not bounded for $\alpha \to 0$ and $\hat{\mu}(0, H) = \mathcal{O}(\sqrt{H})$, only.

Proof of Theorem 2.2 and Corollary 2.3. The *global* errors $\epsilon_{y,n}$, $\epsilon_{u,n}$ of the numerical solution $(\tilde{y}, \tilde{u})$ in $(T_n, T_{n+1}]$ may be splitted in terms $e_{y,n}$, $e_{u,n}$ representing the error propagation from $(T_{n-1}, T_n]$ to $(T_n, T_{n+1}]$ and in terms $d_{y,n}$, $d_{u,n}$ that stand for the error contribution of the actual window (*local* errors):

$$\text{(3.12)} \quad \begin{aligned} \epsilon_{y,n} &:= (\tilde{y} - y)|_{(T_n, T_{n+1}]} &= e_{y,n}|_{(T_n, T_{n+1}]} + d_{y,n}|_{(T_n, T_{n+1}]} \, , \\ \epsilon_{u,n} &:= (\tilde{u} - u)|_{(T_n, T_{n+1}]} &= e_{u,n}|_{(T_n, T_{n+1}]} + d_{u,n}|_{(T_n, T_{n+1}]} \end{aligned}$$

with

$$\text{(3.13)} \begin{bmatrix} e_{y,n} \\ e_{u,n} \end{bmatrix} := (\Psi_n^{k_n} \circ \Phi_n) \left(\begin{bmatrix} \tilde{y}|_{(T_{n-1}, T_n]} \\ \tilde{u}|_{(T_{n-1}, T_n]} \end{bmatrix} \right) - (\Psi_n^{k_n} \circ \Phi_n) \left(\begin{bmatrix} y|_{(T_{n-1}, T_n]} \\ u|_{(T_{n-1}, T_n]} \end{bmatrix} \right) ,$$

$$\left[\begin{array}{c} d_{y,n} \\ d_{u,n} \end{array}\right] := \Psi_n^{k_n}\left(\left[\begin{array}{c} \Phi_{y,n}(y|_{(T_{n-1},T_n]}) \\ \Phi_{u,n}(u|_{(T_{n-1},T_n]}) \end{array}\right]\right) - \Psi_n^{k_n}\left(\left[\begin{array}{c} y|_{[T_n,T_{n+1}]} \\ u|_{[T_n,T_{n+1}]} \end{array}\right]\right). \tag{3.14}$$

The second term on the right hand side of (3.14) is equal to $(y,u)|_{[T_n,T_{n+1}]}$ since the analytical solution is a fixed point of Ψ_n. Note that $\epsilon_{.,n}$ are the global errors on $(T_n, T_{n+1}]$ but $e_{.,n}$, $d_{.,n}$ are defined on $[T_n, T_{n+1}]$.

The proof is organized as follows: applying Lemma 3.2 with suitable arguments (Y,U) and $(\hat{Y},\hat{U})$ we get estimates for $e_{.,n}$ and $d_{.,n}$ (parts (i) and (ii) of the proof). In part (iii) these estimates are combined to derive bounds for the global errors $\epsilon_{.,n}$. Here we assume that the numerical solution $(\tilde{y},\tilde{u})$ remains close to the analytical solution (y,u) throughout integration, i.e., for all $m \geq 0$ with $mH \leq T_e$ the global errors are bounded by

$$\|\epsilon_{y,m}\| + \|\epsilon_{u,m}\| \leq \gamma \tag{3.15}$$

with a sufficiently small constant $\gamma > 0$ that is independent of H and α. In part (iv) of the proof it will be shown by induction that (3.15) is always satisfied if $H_0 > 0$ in Theorem 2.2 is sufficiently small. The proof is completed proving the error bound (2.10) of Theorem 2.2 (part (v)).

(i) With

$$\left[\begin{array}{c} \hat{Y} \\ \hat{U} \end{array}\right] = \Phi_n\left(\left[\begin{array}{c} \tilde{y}|_{(T_{n-1},T_n]} \\ \tilde{u}|_{(T_{n-1},T_n]} \end{array}\right]\right), \qquad \left[\begin{array}{c} Y \\ U \end{array}\right] = \Phi_n\left(\left[\begin{array}{c} y|_{(T_{n-1},T_n]} \\ u|_{(T_{n-1},T_n]} \end{array}\right]\right)$$

we have $\Delta_y^{(k_n)} = e_{y,n}$, $\Delta_u^{(k_n)} = e_{u,n}$ in (3.6). The arguments (Y,U) and $(\hat{Y},\hat{U})$ satisfy $\|\hat{Y}(T_n) - Y(T_n)\| \leq \|\tilde{y} - y\|_{(T_{n-1},T_n]}$ and

$$\begin{aligned} \|\hat{Y} - Y\| &\leq L \cdot \|\tilde{y} - y\|_{(T_{n-1},T_n]} = L\|\epsilon_{y,n-1}\|, \\ \|\hat{U} - U\| &\leq L \cdot \|\tilde{u} - u\|_{(T_{n-1},T_n]} = L\|\epsilon_{u,n-1}\| \end{aligned}$$

since the operator Φ_n extrapolates continuously from $(T_{n-1},T_n]$ to $[T_n,T_{n+1}]$ and satisfies a uniform Lipschitz condition with constant L. Therefore the estimate

$$\left[\begin{array}{c} \|e_{y,n}\|_{[T_n,T_{n+1}]} \\ \|e_{u,n}\|_{[T_n,T_{n+1}]} \end{array}\right] \leq \left[\begin{array}{cc} 1 + C_1^* H & C_1^* H \\ C_1^* & \alpha_n^* \end{array}\right] \cdot \left[\begin{array}{c} \|\epsilon_{y,n-1}\|_{(T_{n-1},T_n]} \\ \|\epsilon_{u,n-1}\|_{(T_{n-1},T_n]} \end{array}\right] \tag{3.16}$$

with constants $C_1^* > 0$ and $\alpha_n^* := L(\hat{\mu}^{k_n} + (\hat{\mu} - \hat{\alpha})^{k_n})$ follows from (3.6) and $\hat{\mu} < 1$.

(ii) With

$$\hat{Y} = \Phi_{y,n}(y|_{(T_{n-1},T_n]}), \qquad \hat{U} = \Phi_{u,n}(u|_{(T_{n-1},T_n]}), \qquad (Y,U) = (y,u)|_{[T_n,T_{n+1}]}$$

we have $\Delta_y^{(k_n)} = d_{y,n}$, $\Delta_u^{(k_n)} = d_{u,n}$ and

$$\hat{Y} - Y = \delta y_n, \qquad \hat{U} - U = \delta u_n, \qquad \hat{Y}(T_n) - Y(T_n) = 0$$

such that (3.6) gives a bound for the local errors $d_{y,n}$, $d_{u,n}$ in terms of the extrapolation errors δy_n, δu_n:

$$(3.17)\quad \|d_{y,n}\| + H\|d_{u,n}\| \le C_2^* H\delta_n \;\text{ with }\; \delta_n := \hat{\mu}^{\max(0,k_n-2)}\|\delta y_n\| + \hat{\mu}^{k_n-1}\|\delta u_n\|$$

and a constant $C_2^* > 0$ that is independent of H, α, and k_n.

(iii) Summarizing (3.12)–(3.14) and (3.16)–(3.17) we end up with

$$(3.18)\quad \begin{bmatrix} \|\epsilon_{y,n}\| \\ \|\epsilon_{u,n}\| \end{bmatrix} \le \begin{bmatrix} 1 + C_1^* H & C_1^* H \\ C_1^* & \alpha^* \end{bmatrix} \cdot \begin{bmatrix} \|\epsilon_{y,n-1}\| \\ \|\epsilon_{u,n-1}\| \end{bmatrix} + \begin{bmatrix} C_2^* H\delta_n \\ C_2^* \delta_n \end{bmatrix}$$

for all $n \ge 0$ with $nH \le T_e$, $\alpha^* := \max_{m \le n} \alpha_m^*$ (set $\|\epsilon_{y,-1}\| = \|\epsilon_{u,-1}\| := 0$ for a unified notation). Coupled error recursions of the form (3.18) are well known from the convergence analysis of one-step methods for index-1 systems [5]; see also [12, Lemma VI.3.9]. If the contractivity condition $\alpha^* < 1$ is satisfied then the error recursion (3.18) is stable and results in

$$(3.19)\qquad \|\epsilon_{y,n}\| + \|\epsilon_{u,n}\| \le C^* \cdot \max_{0 \le m < n} \delta_m$$

with a constant $C^* > 0$ that is independent of n and H. We have

$$\alpha_m^* = L((\hat{\alpha} + \mathcal{O}(\sqrt{H}))^{k_m} + \mathcal{O}(\sqrt{H}^{k_m}))\,, \quad \hat{\alpha} = \alpha + \mathcal{O}(\gamma) + \mathcal{O}(H)\,, \quad L\alpha^{k_m} \le \bar{\alpha} < 1$$

such that $\alpha^* < 1$ is guaranteed if γ and H are sufficiently small.

(iv) Because of $\delta_m = \mathcal{O}(H)$ the right hand side of (3.19) satisfies $C^* \max_m \delta_m \le \gamma$ for all $H \in (0, H_0]$ if $H_0 > 0$ is sufficiently small. Therefore (3.19) may be used to prove (3.15) by induction proceeding step by step from $m = 0$ to $m = n$ with $nH \le T_e$.

(v) The constants $\hat{\alpha}$ and $\hat{\mu}$ in Lemmas 3.1 and 3.2 are given by

$$\hat{\alpha} = \alpha + \mathcal{O}(1) \cdot \max_{0 \le m < n} (\|\epsilon_{y,m}\| + \|\epsilon_{u,m}\|) + \mathcal{O}(H) = \alpha + \mathcal{O}(H)\,,$$

$$\hat{\mu}(\hat{\alpha}, H) = \hat{\mu}(\alpha + \mathcal{O}(H), H) \le \alpha + C_0 \frac{H}{\alpha + \sqrt{H}} = \mu(\alpha, H)$$

with a suitable constant $C_0 > 0$ that is independent of α and H. Inserting this expression for $\hat{\mu}$ in (3.17) and (3.19) the proof is completed. □

4 Application to quasilinear and higher index problems.

The dynamic iteration method (2.2)–(2.3) that has been introduced for semi-explicit index-1 systems (1.1) may be applied as well to quasilinear and/or higher index systems. As long as these more complicated problems may be transformed analytically to the form (1.1) the results of the convergence analysis in Section 2 and Section 3 may be carried over straightforwardly to the quasilinear higher index case since the dynamic iteration method is invariant w.r.t. such analytical transformations. In the present section this will be illustrated by applied problems from electrical and mechanical engineering.

4.1 Electrical circuit simulation.

The (static) circuit partitioning created by the design process serves as a basis for a user specified hierarchical subcircuit structure. Applying domain decomposition methods to large, digital MOS integrated circuit simulation, this system can be decoupled by introducing voltage and/or current sources as coupling units at the boundaries [22]. Using classical Modified Nodal Analysis (MNA), the first approach yields network equations of the quasilinear-implicit type

$$C_i(y_i) \cdot \dot{y}_i + \varphi_i(y) + s_i(t) - P_i^T u = 0\,,\ (i = 1, \ldots, r) \tag{4.1a}$$

for r subcircuits coupled by n_u linear equations

$$0 = \sum_{i=1}^{r} P_i \cdot y_i\,. \tag{4.1b}$$

The state variable y_i contains the vector of node potentials of the i-th subcircuit. Corresponding to n_u coupling conditions (4.1b), the vector u denotes the branch currents through current sources that link the subcircuits. The network equations (4.1a) for each subcircuit consist of four parts. First, the dynamic currents are given by $C_i(y_i) \cdot \dot{y}_i$ due to capacitances, where we restrict ourselves to the case of regular capacitance matrices C_i along the solution and assume that subcircuits are not connected by capacitive paths. The static contribution enters the equations via the nonlinear functions $\varphi_i(y)$ that may depend on all node potentials via controlled sources. Time-dependent input voltages are described by the function $s_i(t)$. Finally, the matrices $P_i \in \{-1, 0, 1\}^{n_u \times n_{y_i}}$ assemble the coupling branch currents u at the boundary nodes.

The automatic modelling approach results in quasilinear-implicit coupled network equations (4.1) that are of index 2. Analytically, these equations can be transformed into the semi-explicit form (1.1) introducing $z_i := \dot{y}_i$ and differentiating the coupling conditions (4.1b) once w.r.t. t:

$$f_i(y, z_i) := z_i\,, \quad h_i(y, z_i, u) := C_i(y_i) z_i + \varphi_i(y) + s_i(t) - P_i^T u \tag{4.2a}$$

with coupling conditions

$$g(y, z) := \sum_{i=1}^{r} P_i \cdot z_i\,. \tag{4.2b}$$

Restricting ourselves as before to the special case $r = 2$ a dynamic iteration method for (4.1) is defined by

$$\begin{array}{rcl} C_1(\tilde{y}_{1,n}^{(k)})\dot{\tilde{y}}_{1,n}^{(k)} + \varphi_1(\tilde{y}_{1,n}^{(k)}, \tilde{y}_{2,n}^{(k-1)}) + s_1(t) - P_1^T \tilde{u}_n^{(k-1)} & = & 0\,, \\ C_2(\tilde{y}_{2,n}^{(k)})\dot{\tilde{y}}_{2,n}^{(k)} + \varphi_2(\tilde{y}_{1,n}^{(k)}, \tilde{y}_{2,n}^{(k)}) \quad + s_2(t) - P_2^T \tilde{u}_n^{(k)} & = & 0\,, \\ P_1 \tilde{y}_{1,n}^{(k)} + P_2 \tilde{y}_{2,n}^{(k)} & = & 0\,. \end{array} \tag{4.3}$$

If the coupling conditions $\sum_i P_i \tilde{y}_{i,n}^{(k)} = 0$ are differentiated w.r.t. t then (4.3) gets the form (2.3) with $\tilde{z}_{i,n}^{(k)} := \dot{\tilde{y}}_{i,n}^{(k)}$ and the notations (4.2). Therefore Theorem 2.2

may be applied also in this quasilinear index-2 case. The contractivity constant is given by

$$\alpha = \max_{t\in[0,T]} \|(P_2 C_2^{-1}(y_2(t))P_2^T)^{-1}(P_1 C_1^{-1}(y_1(t))P_1^T)\| .$$

It depends only on the capacitances (as functions of node potentials y_i) which are connected to the boundaries of both subcircuits. Similar to the method of Section 2.4 the convergence of (4.3) may always be enforced by preconditioning.

REMARK 4.1. The case of singular capacitance matrices $C_i(y_i)$ in the subcircuits is substantially more complicated and not covered by the convergence analysis of Section 3. If some of the capacitance matrices $C_i(y_i)$ are singular then the preconditioned dynamic iteration method may even fail to converge for any choice of the preconditioner; see [8, 9] for a more detailed discussion.

4.2 Multibody system dynamics.

Mechanical multibody system (MBS) models are frequently used in robotics, vehicle system dynamics, and biomechanics [20]. The MBS model setup for complex mechanical systems is commonly based on a splitting into substructures that are in a first step modelled separately. In the second (and final) step these substructure models are coupled by force terms or by (holonomic) constraints to get the MBS model for the overall system (see Section 5 below for a typical example).

The equations of motion are obtained from the equilibria of forces and momenta in the MBS [20]. They have the form

$$\begin{aligned} M_i(q_i)\ddot{q}_i &= \varphi_i(q,\dot{q}) - G_i^T(q)\lambda\,, \quad i=1,\dots,r\,, \\ 0 &= \gamma(q) \end{aligned} \tag{4.4}$$

with q_i denoting the position coordinates of the i-th substructure that are summarized in $q = (q_1,\dots,q_r)^T$. The second order differential equations for the individual substructures are coupled by constraints $\gamma(q) = 0$ that result in constraint forces $-G_i^T\lambda$ with $G_i := \partial\gamma/\partial q_i$ and Lagrangian multipliers λ. All other forces are summarized in the terms $\varphi_i(q,\dot{q})$. The generalized mass matrices M_i are symmetric and positive definite. Furthermore we suppose that the matrices G_i have full rank.

Equations (4.4) form a differential-algebraic system of index 3 that may be transformed by index reduction to an analytically equivalent index-1 system [12, Section VII.1]. If the constraints $\gamma(q) = 0$ are differentiated twice w.r.t. t then (4.4) gets the form (1.1) with velocities $v_i := \dot{q}_i$, accelerations $a_i := \dot{v}_i = \ddot{q}_i$, differential components $y_i := (q_i, v_i)^T$, algebraic components $z_i := a_i$, $u := \lambda$, and

$$\begin{aligned} f_i := \begin{bmatrix} v_i \\ a_i \end{bmatrix}, \qquad & h_i := M_i(q_i)a_i - \varphi_i(q,v) + G_i^T(q)\lambda\,, \\ & g := \sum_{i=1}^{r} G_i(q)a_i + \gamma_{qq}(v,v)\,. \end{aligned} \tag{4.5}$$

The example in Section 5 below illustrates that distributed time integration methods for (4.4) are especially attractive if the system consists of substructures with different solution behaviour. In the special case $r = 2$ such a method is given by the dynamic iteration scheme

$$\begin{aligned}
&\text{(4.6a)} \quad M_1(\tilde{q}_{1,n}^{(k)})\ddot{\tilde{q}}_{1,n}^{(k)} = \varphi_1(\tilde{q}_{1,n}^{(k)}, \tilde{q}_{2,n}^{(k-1)}, \dot{\tilde{q}}_{1,n}^{(k)}, \dot{\tilde{q}}_{2,n}^{(k-1)}) - G_1^T(\tilde{q}_{1,n}^{(k)}, \tilde{q}_{2,n}^{(k-1)})\tilde{\lambda}_n^{(k-1)}\,,\\
&\text{(4.6b)} \quad M_2(\tilde{q}_{2,n}^{(k)})\ddot{\tilde{q}}_{2,n}^{(k)} = \varphi_2(\tilde{q}_{1,n}^{(k)}, \tilde{q}_{2,n}^{(k)}, \dot{\tilde{q}}_{1,n}^{(k)}, \dot{\tilde{q}}_{2,n}^{(k)}) - G_2^T(\tilde{q}_{1,n}^{(k)}, \tilde{q}_{2,n}^{(k)})\tilde{\lambda}_n^{(k)}\,,\\
&\text{(4.6c)} \quad 0 = \gamma(\tilde{q}_n^{(k)})\,.
\end{aligned}$$

As in Section 4.1 Theorem 2.2 may also be applied to study the convergence of (4.6). Since the contractivity constant is given by

$$\alpha := \max_{t\in[0,T]} \|[(G_2M_2^{-1}G_2^T)^{-1}(G_1M_1^{-1}G_1^T)](q(t))\|$$

the contractivity condition $\alpha < 1$ gives a rule of thumb: method (4.6) may be expected to work well if and only if the substructure with larger mass is selected as first subsystem ("$\|M_2M_1^{-1}\| < 1$").

Similar to Definition 2.2 we substitute for the preconditioned version of (4.6) the iterate $\tilde{\lambda}_n^{(k)}$ in (4.6) by an auxiliary function

$$\Lambda_n^{(k)}(t) = \big(I - A(t)\big)\tilde{\lambda}_n^{(k)}(t) + A(t)\tilde{\lambda}_n^{(k-1)}(t)$$

with a preconditioner $A(t)$ [1]. Because of the special structure of (4.4) there is an efficient implementation of the optimal preconditioner (2.14). After some straightforward transformations we get

$$\text{(4.7)} \qquad \begin{bmatrix} M_1 & 0 & G_1^T \\ 0 & M_2 & G_2^T \\ G_1 & G_2 & 0 \end{bmatrix} \begin{bmatrix} \bar{a}_{1,n} \\ \bar{a}_{2,n} \\ \tilde{\lambda}_n^{(k)} \end{bmatrix} = \begin{bmatrix} G_1^T\tilde{\lambda}_n^{(k-1)} \\ G_2^T\Lambda_n^{(k)} \\ 0 \end{bmatrix}$$

with auxiliary vectors $\bar{a}_{i,n}$. Both the coefficients and the right hand side of (4.7) have already been evaluated before such that the additional effort of the preconditioned method is restricted to the solution of this linear system.

Example 4.1. To simulate the dynamic interaction of high-speed trains and bridges Duffek [6] combines an MBS model for the railway vehicle with an elastic beam model for the bridge. This coupled problem is integrated by a method that may be interpreted as a fixed stepsize multi-rate discretization of (4.6) with the beam model as first subsystem ($k_n = 1$, no preconditioning).
The equations of motion for this elastic structure are linear, they have eigenvalues with large imaginary parts. Therefore a special adapted integration method ("guideway operator") is applied with a very small stepsize to guarantee a stable but nevertheless efficient integration of (4.6a). Much larger stepsizes may be used for the MBS model in the second subsystem (4.6b)–(4.6c) that is integrated by a half-explicit Runge–Kutta method.

This distributed time integration method was implemented in an industrial simulation package. It works quite well such that (from the engineer's point of view) there has been no need for a convergence analysis. But a closer look shows that for the technical parameters being given in [6] the contractivity condition is satisfied with $\alpha \approx 0.6$.

5 Numerical tests.

Dynamic iteration methods require the solution of initial value problems for the subsystems; see (2.3). If these initial value problems cannot be solved analytically then additional discretization errors are introduced in the distributed time integration. The numerical methods that are used in the subsystems may even influence the contractivity constant that gets the general form $\alpha_{\Delta} \cdot \alpha$ with α of (2.8) and a method dependent coefficient α_{Δ}. A detailed analysis of these effects is, however, out of the scope of the present paper.

The practical importance of the convergence analysis of Section 2 is illustrated by numerical tests for a rather complex problem from mechanical engineering. In [21] we consider the dynamical interaction between the pantograph of a high-speed train and the catenary; see Figure 5.1. This coupled system is modelled combining a MBS model for the substructure "pantograph" with a beam model for the substructure "catenary" that is semi-discretized in space by finite differences. Then the equations of motion have the form (4.4) with $n_{q_{\mathrm{p}}}$ position coordinates q_{p} for the pantograph and $n_{q_{\mathrm{c}}}$ position coordinates q_{c} for the (semi-discretized) catenary. The scalar constraint $\gamma(q_{\mathrm{p}}, q_{\mathrm{c}}) = 0$ guarantees permanent contact between pantograph head and catenary.

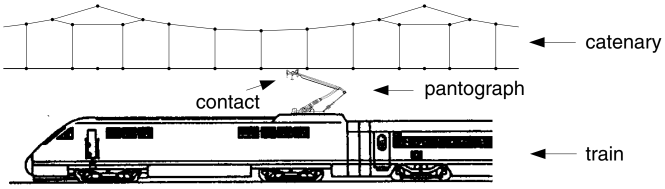

Figure 5.1: System pantograph/catenary; see [21, Figure 1].

The numerical solution of the coupled equations (4.4) is non-trivial since the high-speed motion of the pantograph head causes nearly undamped high frequency oscillations in the catenary.

In principle, (4.4) could be integrated by any standard method like, e.g., the half-explicit Störmer method of [2] that will be used as reference method throughout this section. With this straightforward approach one and the same time stepsize is used in both subsystems and the pantograph substructure causes more than 90% of the overall computing time since realistic MBS models for the technical system pantograph are highly nonlinear and stiff and have a few hundred degrees of freedom [21].

There is a great potential to save computing time by a multi-rate method since the stepsize τ for the catenary subsystem is strongly restricted for stability

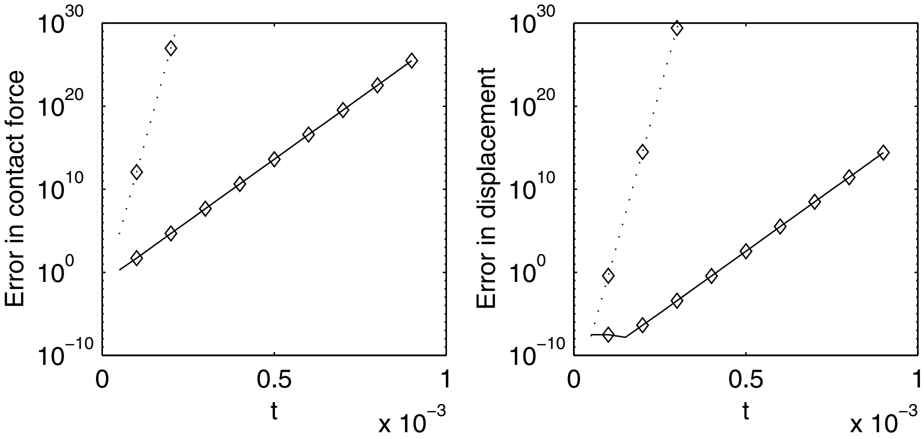

Figure 5.2: Benchmark "Pantograph": without preconditioning the distributed time integration method is unstable if the catenary is selected as first subsystem (solid lines: $\tau = 5.0 \cdot 10^{-5}$; dotted lines: $\tau = 1.0 \cdot 10^{-5}$).

reasons and much larger stepsizes τ^+ may be used for the pantograph subsystem. The stability bound for τ results from large eigenvalues close to the imaginary axis. It has the form $\tau \leq c \cdot \Delta_x^2$ with Δ_x denoting the mesh width in the space discretization of the beam model [2].

In the numerical tests we restrict ourselves to a simplified benchmark problem with $n_{q_\mathrm{p}} = 2$ and $n_{q_\mathrm{c}} = 998$; see [2] for the complete model data. In this benchmark the strongly simplified pantograph is moved with constant speed $v_\mathrm{p} = 32.0\,\mathrm{m/s}$ along a catenary with the (fictitious) length $L = 20.0\,\mathrm{m}$.

We consider several distributed time integration methods that are all based on the dynamic iteration method (4.6) with fixed window sizes $H = \tau^+ \geq \tau$, fixed numbers of iteration steps $k_n = \bar{k}$ and constant extrapolated data as initial guesses:

$$\tilde{q}_{2,n}^{(0)}(T) = \tilde{q}_2(T_n)\,, \quad \tilde{v}_{2,n}^{(0)}(T) = \tilde{v}_2(T_n)\,, \quad \tilde{a}_{2,n}^{(0)}(T) = \tilde{a}_2(T_n)\,, \quad \tilde{\lambda}_n^{(0)}(T) = \tilde{\lambda}(T_n)\,,$$

$(\,T \in [T_n, T_{n+1}]\,)$. The initial value problems for the subproblems (4.6a) and (4.6b)–(4.6c) are integrated by half-explicit methods applied to the stabilized index-1 formulation of (4.6) with fixed stepsizes τ and $\tau^+ \geq \tau$ that are chosen such that τ is close to the stability bound $c \cdot \Delta_x^2 \approx 6.6 \cdot 10^{-5}$; see [12] for a general introduction to half-explicit methods and [2] for a more detailed discussion of the application to the stabilized index-1 formulation of (4.6).

In view of the positive results of Duffek [6] (see Example 4.1) it seems to be reasonable to select the elastic structure as first subsystem. Figure 5.2 shows the exponential instability of this approach. Both subsystems have been integrated by the 2nd order Störmer method with stepsizes $\tau = \tau^+ = H$. The results get even worse if τ is decreased. This behaviour is in perfect agreement with the

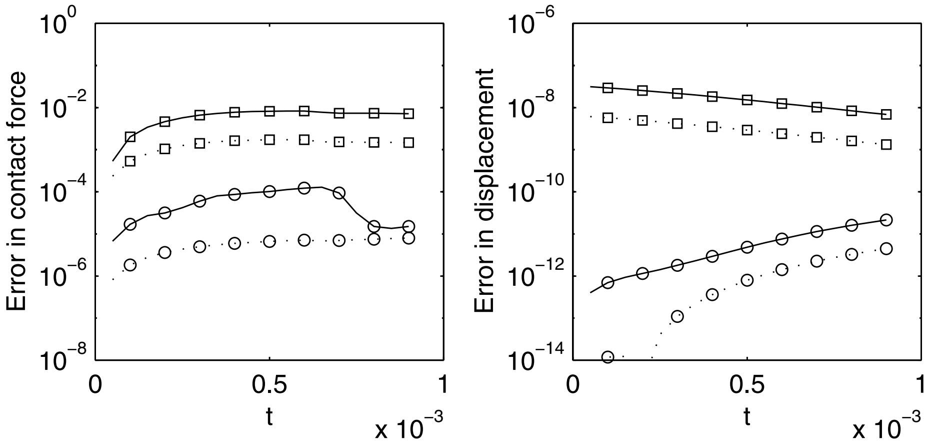

Figure 5.3: Benchmark "Pantograph": the instability of the distributed time integration method may be fixed by preconditioning ("□") or by changing the order of the subsystems ("○") (solid lines: $\tau = 5.0 \cdot 10^{-5}$; dotted lines: $\tau = 1.0 \cdot 10^{-5}$).

error analysis of Section 2.3 since the contractivity constant is

$$\alpha \approx |(0.11)^{-1} \cdot 3.44\,| = 31.\overline{27}$$

and an error of size $\mathcal{O}(\alpha^n)$ may be observed (see also (3.18)). The method dependent coefficient is $\alpha_\vartriangle = 1$. The approach of Duffek fails in this application since the mass of the catenary is several orders of magnitude smaller than the mass of a railway bridge.

Figure 5.3 shows that convergence may be achieved using the optimal preconditioner (2.14). Alternatively, the order of the subsystems may be changed such that now the pantograph is selected as first subsystem ($\alpha \approx 0.11/3.44 < 0.04$). In both cases the distributed time integration method is stable and the error gets smaller if τ is reduced. The relative error in the vertical displacement of the pantograph head (= error in q) is smaller than the relative error in the contact force (= error in λ). Furthermore, this example illustrates that the order of the subsystems does not only influence the stability of the distributed time integration method but also its error constants, even if preconditioning is used.

The results of Figures 5.4–5.6 have been obtained without preconditioning and with the pantograph as first subsystem. The catenary subsystem is always integrated by the 2nd order Störmer method with $\tau = 5.0 \cdot 10^{-5}$. Larger stepsizes τ^+ are used for the pantograph subsystem such that this system has to be integrated with a dense output formula to provide the necessary data at the intermediate grid points $T_n + \tau$, $T_n + 2\tau$, $\ldots$, $T_n + \tau^+ - \tau$. The 2nd order Störmer method gives a dense output of order 2 for $\tilde{q}_{1,n}^{(k)}$, $\dot{\tilde{q}}_{1,n}^{(k)}$ and of order 1 for $\ddot{\tilde{q}}_{1,n}^{(k)}$. The classical 4th order Runge–Kutta method gives a dense output of order 3 for $\tilde{q}_{1,n}^{(k)}$, $\dot{\tilde{q}}_{1,n}^{(k)}$ and of order 2 for $\ddot{\tilde{q}}_{1,n}^{(k)}$.

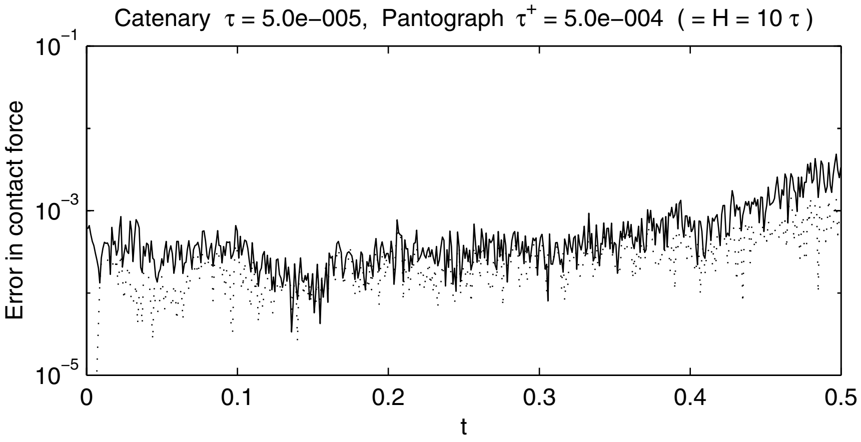

Figure 5.4: Benchmark "Pantograph": error of the distributed time integration method with 2nd order Störmer method for both subsystems, $k_n = 1$, and stepsizes $\tau = 5.0 \cdot 10^{-5}$ and $\tau^+ = 10\tau$ (solid line) and error of the reference method, 2nd order Störmer method with stepsize $\tau = 5.0 \cdot 10^{-5}$ applied to the overall system (4.4)(dotted line).

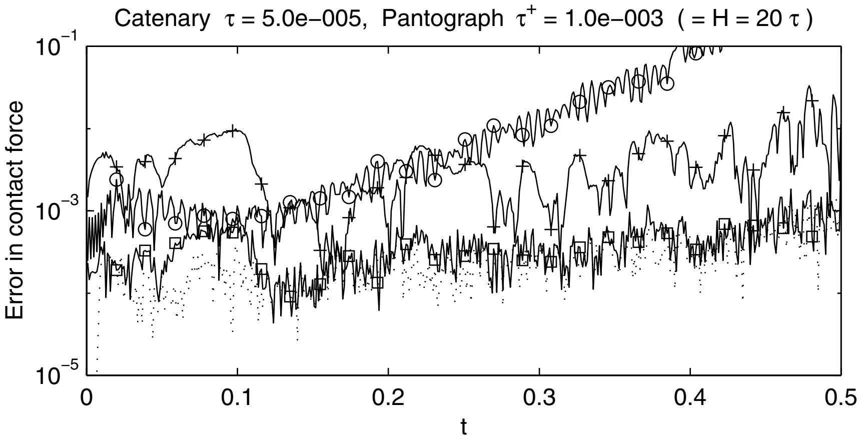

Figure 5.5: Benchmark "Pantograph": the solid lines show the errors of several distributed time integration methods with stepsizes $\tau = 5.0 \cdot 10^{-5}$ and $\tau^+ = 20\tau$. "$+$" ($k_n = 1$) and "$\Box$" ($k_n = 2$): combination of 4th order Runge–Kutta and 2nd order Störmer method, "$\circ$"; 2nd order Störmer method for both subsystems ($k_n = 1$). The dotted line shows the error of the reference method.

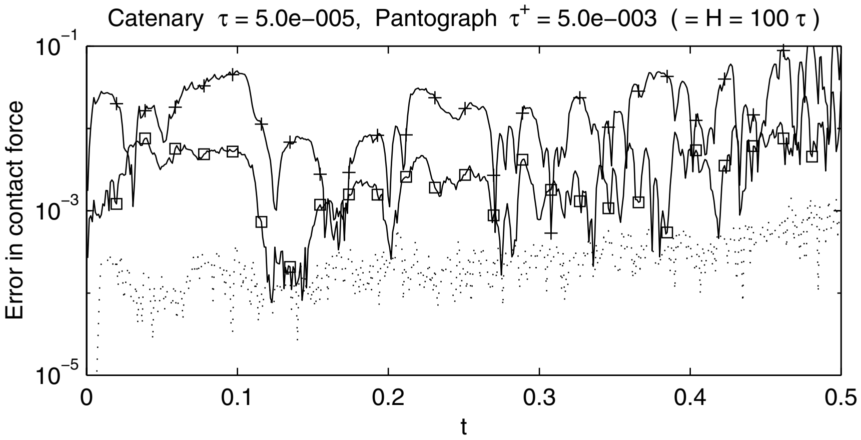

Figure 5.6: Benchmark "Pantograph": error of the distributed time integration method with a combination of 4th order Runge–Kutta and 2nd order Störmer method and stepsizes $\tau = 5.0 \cdot 10^{-5}$ and $\tau^+ = 100\tau$ (solid lines, "+": $k_n = 1$, "□": $k_n = 2$) and error of the reference method (dotted line).

The numerical results of these distributed time integration methods and of the reference method of [2] with $\tau = 5.0 \cdot 10^{-5}$ have been compared with a reference solution that was computed with the reference method and the very small stepsize $\tau = 1.0 \cdot 10^{-5}$. Figures 5.4–5.6 show the (relative) error in λ that is of size 10^{-3} for the reference method (dotted lines). From a practical point of view errors up to 10^{-2} are acceptable since the model error is in the range 5–10%.

Using the 2nd order Störmer method with $k_n = 1$ the stepsize $\tau^+ = 10\tau$ gives excellent results (Figure 5.4) but the error increases rapidly if $\tau^+ = 20\tau$ (Figure 5.5). In both cases the results remain nearly unchanged if $k_n > 1$ iterations are performed. The discretization error of the 2nd order method dominates and the error of the dynamic iteration is negligible.

The error is decreased substantially using the 4th order Runge–Kutta method for the pantograph subsystem. If $k_n = 2$ then the error is of size 10^{-3} if $\tau^+ = 20\tau$ (Figure 5.5) and of size 10^{-2} if $\tau^+ = 100\tau$ (Figure 5.6). There is no further error reduction if $k_n > 2$. Figures 5.5 and 5.6 illustrate that the iteration error is clearly dominating if $k_n = 1$ such that it really pays to perform a second iteration step in (4.6). If the 4th order method is used then the dynamic iteration method with $k_n \geq 2$ is superior to a standard multi-rate method (that may be interpreted as a discretization of (4.6) with $k_n = 1$).

The results for the benchmark problem show that the overall effort for the solution of coupled differential-algebraic systems may be reduced substantially using suitable distributed time integration methods.

6 Summary.

Considering dynamic iteration, we analysed in the present paper the convergence of distributed time integration methods for coupled differential-algebraic systems of index one. For an efficient practical implementation of the dynamic iteration method the time interval of interest is split into windows. We considered a finite number k of iteration steps in each window and studied the error propagation from window to window. This analysis covers also well known non-iterative techniques like multi-rate methods and multi-method approaches.

It has been known from the literature that a contractivity condition $\alpha < 1$ is necessary for stability and convergence of dynamic iteration methods for coupled differential-algebraic systems. As a new result, a second contractivity condition $L\alpha^k < 1$ has to be introduced in order to guarantee a stable error propagation from window to window. Furthermore, we showed that both contractivity conditions may always be achieved by a suitable splitting w.r.t. the algebraic solution components. To allow a comparison between different splittings all error estimates have been formulated with constants independent of $\alpha \in (0, 1)$.

The theoretical results were applied to quasilinear problems from electrical circuit simulation and to index-3 systems in multibody dynamics. Together with the numerical results for the pantograph benchmark, these applications underline the relevance of the theoretical results obtained in the paper also from a practical point of view.

Acknowledgments.

The authors are grateful to B. Leimkuhler (University of Leicester) and P. Spellucci (Technical University Darmstadt) for providing an extensive list of references on waveform relaxation methods in technical applications. Furthermore we thank W. Kortüm (DLR Oberpfaffenhofen) and P. Rentrop (University of Karlsruhe) for their permanent support of our activities.

REFERENCES

1. M. Arnold, *A pre-conditioned method for the dynamical simulation of coupled mechanical multibody systems*, to appear in: Z. Angew. Math. Mech.
2. M. Arnold and B. Simeon, *Pantograph and catenary dynamics: A benchmark problem and its numerical solution*, Appl. Numer. Math., 34 (2000), pp. 345–362.
3. K. Brenan, S. Campbell, and L. Petzold, *Numerical Solution of Initial-value Problems in Differential-algebraic Equations*, 2nd ed., SIAM, Philadelphia, 1996.
4. K. Burrage, Z. Jackiewicz, S. Nørsett, and R. Renaut, *Preconditioning waveform relaxation iterations for differential systems*, BIT, 36 (1996), pp. 54–76.
5. P. Deuflhard, E. Hairer, and J. Zugck, *One-step and extrapolation methods for differential-algebraic systems*, Numer. Math., 51 (1987), pp. 501–516.
6. W. Duffek, *Ein Fahrbahnmodell zur Simulation der dynamischen Wechselwirkung zwischen Fahrzeug und Fahrweg*, Tech. Report IB 515–91–18, DLR, D-5000 Köln 90, 1991.
7. C. W. Gear and D. R. Wells, *Multirate linear multistep methods*, BIT, 24 (1984), pp. 484–502.

8. G. Gristede, C. Zukowski, and A. Ruehli, *Measuring error propagation in waveform relaxation analysis*, IEEE Trans. Circuits Systems Fund. Theory Appl., 46 (1999), pp. 337–348.
9. M. Günther and M. Arnold, *Coupled simulation of partitioned differential-algebraic network models*, in preparation.
10. M. Günther and P. Rentrop, *Multirate ROW methods and latency of electric circiuts*, Appl. Numer. Math., 13 (1993), pp. 83–102.
11. E. Hairer, S. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations. I. Nonstiff Problems*, 2nd ed, Springer-Verlag, Berlin, 1993.
12. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations. II. Stiff and Differential-Algebraic Problems*, 2nd ed., Springer-Verlag, Berlin, 1996.
13. M. Hoschek, P. Rentrop, and Y. Wagner, *Network approach and differential-algebraic systems in technical applications*, Surveys Math. Indust., 9 (1999), pp. 49–75.
14. Z. Jackiewicz and M. Kwapisz, *Convergence of waveform relaxation methods for differential-algebraic systems*, SIAM J. Numer. Anal., 33 (1996), pp. 2303–2317.
15. R. Kübler and W. Schiehlen, *Two methods of simulator coupling*, Math. Comput. Modelling Dynamical Syst., 6 (2000), pp. 93–113.
16. E. Lelarasmee, A. Ruehli, and A. Sangiovanni-Vincentelli, *The waveform relaxation method for time domain analysis of large scale integrated circuits*, IEEE Trans. CAD of IC and Syst., 1 (1982), pp. 131–145.
17. R. Lewis, P. Bettess, and E. Hinton, eds., *Numerical Methods in Coupled Systems*, Wiley Series in Numerical Methods in Engineering, Wiley, Chichester, 1984.
18. U. Miekkala and O. Nevanlinna, *Convergence of dynamic iteration methods for initial value problems*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 459–482.
19. P. Rentrop, *Partitioned Runge–Kutta methods of order four with stepsize control for stiff ordinary differential equations*, Numer. Math., 47 (1985), pp. 545–564.
20. R. Roberson and R. Schwertassek, *Dynamics of Multibody Systems*, Springer-Verlag, Berlin, 1988.
21. A. Veitl and M. Arnold, *Coupled simulation of multibody systems and elastic structures*, in Advances in Computational Multibody Dynamics, J. Ambrósio and W. Schiehlen, eds., IDMEC/IST, Lisbon, Portugal, 1999, pp. 635–644.
22. U. Wever and Q. Zheng, *Parallel transient simulation on workstation clusters*, in Progress in Industrial Mathematics at ECMI 94, H. Neunzert, ed., Wiley & Teubner, Chichester, 1996, pp. 274–284.
23. M. Witting and T. Pröpper, *Cosimulation of electromagnetic fields and electrical networks in the time domain*, Surveys Math. Indust., 9 (1999), pp. 101–116.