Int. J. Appl. Math. Comput. Sci., 2009, Vol. 19, No. 3, 381–397
DOI: 10.2478/v10006-009-0032-4

A NOVEL INTERVAL ARITHMETIC APPROACH FOR SOLVING DIFFERENTIAL-ALGEBRAIC EQUATIONS WITH VALENCIA-IVP

ANDREAS RAUH [*], MICHAEL BRILL [**], CLEMENS GÜNTHER [**]

[*] Chair of Mechatronics
University of Rostock, D-18059 Rostock, Germany
e-mail: andreas.rauh@uni-rostock.de

[**] Institute of Measurement, Control, and Microtechnology
University of Ulm, D-89069 Ulm, Germany
e-mail: m.brill@gmx.net, guenther.clemens@googlemail.com

The theoretical background and the implementation of a new interval arithmetic approach for solving sets of differential-algebraic equations (DAEs) are presented. The proposed approach computes guaranteed enclosures of all reachable states of dynamical systems described by sets of DAEs with uncertainties in both initial conditions and system parameters. The algorithm is based on VALENCIA-IVP, which has been developed recently for the computation of verified enclosures of the solution sets of initial value problems for ordinary differential equations. For the application to DAEs, VALENCIA-IVP has been extended by an interval Newton technique to solve nonlinear algebraic equations in a guaranteed way. In addition to verified simulation of initial value problems for DAE systems, the developed approach is applicable to the verified solution of the so-called inverse control problems. In this case, guaranteed enclosures for valid input signals of dynamical systems are determined such that their corresponding outputs are consistent with prescribed time-dependent functions. Simulation results demonstrating the potential of VALENCIA-IVP for solving DAEs in technical applications conclude this paper. The selected application scenarios point out relations to other existing verified simulation techniques for dynamical systems as well as directions for future research.

Keywords: ordinary differential equations, differential-algebraic equations, VALENCIA-IVP, verified simulation, inverse control problems.

1. Introduction

To describe and analyze the dynamics of a large number of systems in engineering, physics, and (bio-) chemical as well as (bio-) medical applications, continuous-time mathematical models are commonly used. These continuous-time models can either be given in the form of system models with lumped parameters or distributed parameters.

In this article, the lumped parameter case is studied in detail. Here, state-space models given by sets of explicit ordinary differential equations (ODEs) as well as sets of differential-algebraic equations (DAEs) are considered. The case of distributed parameters which requires descriptions including partial differential equations is not studied in the following.

In contrast to mathematical system models described by sets of ODEs, descriptions using DAEs are often more natural, since they allow expressing directly both differential and algebraic relations between the relevant variables. Especially algebraic equations can be used to formulate constraints on the system dynamics. Furthermore, they are applicable to formulate common problems in control engineering involving optimal trajectory planning for both open-loop and closed-loop control systems.

During the last decade, interval arithmetic techniques have been developed as a means to compute guaranteed enclosures of the trajectories of dynamical systems under consideration of bounded uncertainties of system parameters and initial conditions (Berz and Makino, 1998; Lin and Stadtherr, 2007; Nedialkov, 2007).

Since these techniques allow providing guaranteed state enclosures in a single evaluation of the state equations over the desired time horizon, they are often preferred to non-set-theoretic approaches such as Monte Carlo simulations or grid-based techniques if worst-case state enclosures are of interest. In contrast to interval methods, Monte Carlo simulations or grid-based approaches lead to the necessity for repeated simulations of the same mathematical model with various parameterizations. Although this results in an increased computational effort, no guarantee can be given that worst-case bounds are obtained for the state variables in the case of dynamical systems with bounded uncertainties of parameters and initial conditions.

In this article, a new algorithmic approach based on interval arithmetic is presented to solve sets of DAEs with uncertainties in both initial states and parameters. This approach is based on the verified ODE solver VALENCIA-IVP (`http://www.valencia-ivp.com`) which has been originally developed to determine guaranteed enclosures of the solutions of sets of ODEs (Auer *et al.*, 2008; Rauh *et al.*, 2007a).

In this context, the term *guaranteed* is used interchangeably with the commonly used terms *verified* and *reliable* to denote the fact that the results given by the enclosures of all reachable states are mathematically (and not empirically) proven to be correct. However, naive implementations of interval algorithms can lead to a significant amount of overestimation (Jaulin *et al.*, 2001; Moore, 1966). In the extreme case, this overestimation can lead to meaningless results. To obtain tightest possible bounds of the exact sets of all reachable states, suitable algorithmic techniques have to be implemented in each interval arithmetic approach to prevent the growth of the diameters of the enclosures of the sets of all reachable states over simulation time.

In Section 2, problem formulations for both interval arithmetic simulations of sets of ODEs and DAEs are summarized. Furthermore, possible application scenarios and alternative simulation approaches are discussed. In Section 3, the basic algorithmic techniques of VALENCIA-IVP for the simulation of ODEs and for the reduction of overestimation are highlighted. Extensions for the simulation of dynamical systems including algebraic constraints, which are based on the identification of hidden constraints in DAE systems, are derived in Section 4. In Sections 5–7, an interval Newton technique for the solution of nonlinear algebraic equations is described, implemented, and interfaced with VALENCIA-IVP to compute guaranteed state enclosures for DAEs. In Section 8, the basic properties of these routines are demonstrated for an electrical network as a simple application scenario. Finally, this paper is concluded with an outlook on future research in Section 9.

2. Problem formulation

Mathematical models of continuous-time dynamical systems are usually either formulated as sets of ODEs or DAEs. For both types of system representations, commonly used notations and basic problem formulations are summarized in the following.

2.1. Ordinary differential equations (ODEs). The classical description of a nonlinear continuous-time system is a set of time-varying ODEs

$$\dot{x}(t) = f\left(x(t), t\right) \tag{1}$$

with the state vector

$$x(t) := \begin{bmatrix} x_s(t) \\ p(t) \end{bmatrix} \in \mathbb{R}^{n_x}. \tag{2}$$

This state vector represents a concatenation of the actual system states $x_s(t) \in \mathbb{R}^{n_s}$ and the time-varying system parameters $p(t) \in \mathbb{R}^{n_p}$. From an application-oriented point of view, the two parts of $x(t)$ have to be distinguished since the components $x_s(t)$ can usually be influenced in a systematic way using feedforward or feedback control strategies, while the quantities $p(t)$ are usually not controllable. If the parameters $p(t)$ are not constant, their variation rates are described by additional ODEs

$$\dot{p}\left(t\right) = \Delta p\left(t\right). \tag{3}$$

The function $f : D \rightarrow \mathbb{R}^{n_x}, D \subset \mathbb{R}^{n_s} \times \mathbb{R}^{n_p} \times \mathbb{R}^1$ is the nonlinear state-space representation (Rauh *et al.*, 2007a) of the dynamical system which is assumed to be differentiable with respect to x and t up to the orders that are required by the specific solution approaches, which are applied to solve initial value problems. To account for uncertainties in the state equations (1), initial states $x_0 := x(t_0)$ are considered that are bounded by the guaranteed interval enclosure

$$[x_0] := [\underline{x}_0; \overline{x}_0] = \begin{bmatrix} [\underline{x}_{s0}; \overline{x}_{s0}] \\ [\underline{p}_0; \overline{p}_0] \end{bmatrix}. \tag{4}$$

Furthermore, for time-varying parameters

$$p\left(t\right) \in \left[\underline{p}\left(t\right)\,;\,\overline{p}\left(t\right)\right] \tag{5}$$

the property

$$\Delta p\left(t\right) \in \left[\underline{\Delta p}\left(t\right)\,;\,\overline{\Delta p}\left(t\right)\right] \tag{6}$$

holds.

Guaranteed enclosures of the sets of all reachable states of the system model (1) can be computed with the help of verified ODE solvers such as VALENCIA-IVP. The basic routines for the computation of state enclosures implemented in this solver are summarized in Section 3.

2.2. Differential-algebraic equations (DAEs).

Although mathematical system representations given by sets of explicit ODEs are often easier to solve numerically than sets of implicit ODEs or DAEs, the latter descriptions usually arise in a natural way if mathematical models for dynamical systems in various fields of engineering, physics, and (bio-) chemical as well as (bio-) medical applications are derived and identified.

For example, considering electrical circuits, a mathematical description consisting of a set of coupled ODEs together with algebraic constraints is obtained from direct application of Kirchhoff's voltage and current laws (Chua *et al.*, 1990). Thus, the solution of initial value problems for sets of differential equations which are coupled with algebraic constraints according to

$$\dot{x}(t) = f\big(x(t), y(t), t\big), \tag{7a}$$

$$0 = g\big(x(t), y(t), t\big), \tag{7b}$$

with $f : D \rightarrow \mathbb{R}^{n_x}$ and $g : D \rightarrow \mathbb{R}^{n_y}$, where $D \subset \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^1$ holds, is of special practical importance.

Analogously to sets of explicit ODEs, the goal of this paper is to derive an interval arithmetic solver for initial value problems for sets of DAEs. Using this solver, the influence of bounded parameter uncertainties as well as uncertainties of the initial conditions on the set of all reachable states is analyzed without further symbolic reformulations of the state equations (7a) and (7b).

However, in contrast to ODEs, the solvability of initial value problems for DAEs strongly depends on the consistency of the initial values $x\,(t_0)$ and $y\,(t_0)$. These initial conditions have to fulfill not only the algebraic constraints (7b) but also hidden constraints which can be computed as the Lie derivatives of the algebraic constraints g in the direction of the vector field f.

As in the case of ODEs, the vector $x\,(t)$ of differential state variables contains both constant and time-varying system parameters which cannot be influenced using the control inputs of the dynamical system. The variation rates of these parameters are again described by additional ODEs (3) with (6), which are assumed to be included in (7a) with $\Delta p\,(t) = 0$ and $\Delta p\,(t) \neq 0$, respectively.

2.3. Possible applications.

One reason for extending the verified simulation tool VALENCIA-IVP to handle sets of DAEs is to simplify the steps that have to be performed between mathematical modeling and simulation, i.e., the solution of an initial value problem.

Verified simulations of dynamical system models providing interval enclosures of the sets of all reachable states are a powerful tool for modern techniques in control engineering including the design of optimal and robust control laws (Rauh and Hofer, 2009; Rauh *et al.*, 2009).

As shown, e.g., in (Rauh, 2008), verified solvers for initial value problems are a prerequisite for the implementation of interval observers which allow us to identify regions in the state-space that are consistent with the system dynamics (usually described by ODEs or DAEs) and measured data with bounded uncertainties. Commonly, measured values are only available at discrete points of time. Given that a dynamical system is completely observable, information about all components of its state vector can be reconstructed by measuring some of these components. Classical, well-known non-verified approaches for this task are, e.g., the Luenberger observer and the Kalman filter.

A further application for verified solvers for sets of DAEs is the solution of *inverse control problems*. In this case, control variables for dynamical systems are determined such that specific output signals are consistent with predefined time-dependent functions (Czechowski *et al.*, 2006). Compared with a standard simulation, the role of input and output variables is reversed.

In (Rauh *et al.*, 2008; Rauh and Auer, 2008), VALENCIA-IVP is applied to the verification of the dynamics of mechanical multibody systems using the multibody modeling and simulation environment SMARTMOBILE.

2.4. Further simulation techniques for ODEs and DAEs.

Simulation tools for dynamical system models can be classified into non-verified solvers on the one hand and verified solvers on the other hand. There exists a huge number of different non-verified numerical tools for the computation of approximate solutions of initial value problems for sets of ODEs. These tools are based on explicit as well as implicit single-step and multi-step solvers. Well-known approaches rely, e.g., on Runge-Kutta methods. Ready to use routines can be found, e.g., in the GNU GSL library, which is freely available (Galassi, 2006).

In the case of DAEs, the constraints (7b) can, in general, be eliminated analytically such that only the resulting set of ODEs has to be solved. To avoid this tedious work, routines have been implemented which do not eliminate the algebraic constraints explicitly and which solve the set of DAEs (7) directly.

Numerical approximations for initial value problems for sets of DAEs can be computed by algorithms like DASSL (Petzold, 1982), GAMD (Iavernaro and Mazzia, 1998), GELDA (Kunkel *et al.*, 1997), MEBDFI (Cash and Considine, 1992), PSIDE (de Swart *et al.*, 1998), or RADAU (Hairer *et al.*, 1989; Hairer and Wanner, 1991). The recently developed solver DAETS (Nedialkov and Pryce, 2008) is reported to compute more accurate approximations of the true solutions and to show improved performance for DAE systems with higher differential indices.

Up to now, no general-purpose verified solver for sets of DAEs exists. The only approach that is known to the authors is based on COSY VI (Hoefkens, 2001). However, symbolic reformulations of the set of DAEs to be evaluated are performed in this software such that it does not exactly match the above-mentioned goal to solve DAEs directly.

3. VALENCIA-IVP for the calculation of guaranteed state enclosures for ODEs

VALENCIA-IVP has been developed originally as an initial value problem solver for sets of ODEs with uncertainties in the initial states and parameters (Rauh *et al.*, 2007a; Rauh *et al.*, 2007b). The solution provided by VALENCIA-IVP is an interval enclosure which contains all reachable states of the dynamical system. Therefore, it is guaranteed that—also under consideration of uncertainties—there are definitely no reachable states outside these interval enclosures. Classical non-verified techniques like grid-based approaches or Monte-Carlo methods (Hammersley and Handscomb, 1964) do not guarantee the correctness of the solution.

For initial value problems for ODEs

$$\dot{x}(t) = f(x(t), t) \tag{8}$$

with

$$x(t_0) \in [x_0] = [\underline{x}_0; \overline{x}_0] , \tag{9}$$

the set of all reachable states is enclosed by the time-varying bounds

$$[x_{encl}(t)] := x_{app}(t) + [R(t)] \tag{10}$$

for all points of time $t_0 \leq t \leq t_f$. Without loss of generality, $t_0 = 0$ is assumed in the following.

The interval enclosure in (10) consists of a non-verified approximate solution $x_{app}(t)$ and guaranteed error bounds $[R(t)]$ which are computed iteratively. Generally, arbitrary approximate solutions $x_{app}(t)$ can be used. Currently, VALENCIA-IVP supports the choice between single-step or multiple-step solvers as well as explicit or implicit techniques at this stage. Ready to use algorithms employing, e.g., explicit Runge-Kutta methods or implicit Bulirsch-Stoer methods are implemented in the C++ library GSL (Galassi, 2006).

Theorem 1. *Consider an initial value problem as defined in the equations (8) and (9) with $f : D \mapsto \mathbb{R}^{n_x}$, $D \subset \mathbb{R}^{n_x} \times \mathbb{R}^1$ open, $f \in C^1(D, \mathbb{R}^{n_x})$. Then all reachable states at the point of time t are contained in the interval enclosure (10) if the error bounds $[R(t)]$ are computed by the following two-stage procedure:*

1. Iterative computation of an interval enclosure of all possible time derivatives $\left[\dot{R}(t)\right]$ of the error term by

$$\begin{aligned}\left[\dot{R}^{(\kappa+1)}(t)\right] &= -\dot{x}_{app}(t) + f\left(\left[x_{encl}^{(\kappa)}(t)\right], t\right)\\ &= -\dot{x}_{app}(t) + f\left(x_{app}(t) + \left[R^{(\kappa)}(t)\right], t\right)\\ &=: r\left(\left[R^{(\kappa)}(t)\right], t\right)\end{aligned} \tag{11}$$

for $0 \leq t \leq T$.

This iteration converges to a verified enclosure of $\left[\dot{R}(t)\right]$ if $\left[\dot{R}^{(\kappa+1)}(t)\right] \subseteq \left[\dot{R}^{(\kappa)}(t)\right]$ holds. The iteration (11) is continued until $\left[\dot{R}^{(\kappa+1)}(t)\right] \approx \left[\dot{R}^{(\kappa)}(t)\right]$.

2. Verified integration of $\left[\dot{R}^{(\kappa+1)}(t)\right]$, $0 \leq t \leq T$, $0 \leq T \leq t_f$, with respect to time according to

$$\begin{aligned}\left[R^{(\kappa+1)}(T)\right] &\subseteq \left[R^{(\kappa+1)}(0)\right] + \int_0^T \left[\dot{R}^{(\kappa+1)}(\tau)\right] \mathrm{d}\tau\\ &= \left[R^{(\kappa+1)}(0)\right] + \int_0^T r\left(\left[R^{(\kappa)}(\tau)\right], \tau\right) \mathrm{d}\tau\end{aligned}$$

replaced by the guaranteed bound

$$\begin{aligned}\left[R^{(\kappa+1)}(T)\right] \subseteq{}& \left[R^{(\kappa+1)}(0)\right]\\ &+ T \cdot r\left(\left[R^{(\kappa)}([0\ ;\ T])\right], [0\ ;\ T]\right).\end{aligned} \tag{12}$$

These updated error bounds are required for the evaluation of the formula (11) in the next iteration step. Uncertainties of the initial conditions are accounted for by choosing $[R(0)]$ such that $[x_0] \subseteq x_{app}(0) + [R(0)]$ is fulfilled.

Proof. Using the Picard iteration (Eijgenraam, 1981; Deville *et al.*, 2002),

$$\left[B^{(\kappa+1)}\right] = [x_0] + [0\ ;\ T] \cdot f\left(\left[B^{(\kappa)}\right], [0\ ;\ T]\right), \tag{13}$$

a bounding box $[B]$ of all states which are reachable in the time interval $t \in [0\ ;\ T]$ can be determined according to Banach's fixed-point theorem. Substituting $[x_{encl}]$ as defined in (10) for the bounding box $[B]$ on both sides of (13) leads to the iteration formula

$$\begin{aligned}&\left[x_{encl}^{(\kappa+1)}([0\ ;\ T])\right]\\ &\quad = [x_0] + [0\ ;\ T] \cdot f\left(\left[x_{encl}^{(\kappa)}([0\ ;\ T])\right], [0\ ;\ T]\right).\end{aligned} \tag{14}$$

Let the approximation error $[R(t)]$ in (10) be defined by

$$[R([0\ ;\ T])] := [R(0)] + [0\ ;\ T] \cdot \left[\dot{R}([0\ ;\ T])\right]. \tag{15}$$

Then, the iteration formula (14) is equivalent to

$$\begin{aligned} &x_{app}\left([0\ ;\ T]\right)+\left[R^{(\kappa+1)}\left([0\ ;\ T]\right)\right] \\ &\quad=[x_0]+[0\ ;\ T]\cdot f\left(\left[x_{encl}^{(\kappa)}\left([0\ ;\ T]\right)\right],[0\ ;\ T]\right). \end{aligned} \tag{16}$$

Due to the definition (15), $\left[\dot{R}\left([0\ ;\ T]\right)\right]$ is a conservative interval enclosure of all possible time derivatives of $\left[R\left([0\ ;\ T]\right)\right]$ in the time interval considered. Since the definition (15) holds in each iteration step κ, $\left[\dot{R}^{(\kappa+1)}\left([0\ ;\ T]\right)\right]$ is a guaranteed interval enclosure of all possible time derivatives of $\left[R^{(\kappa+1)}\left([0\ ;\ T]\right)\right]$ in the time interval $[0\ ;\ T]$.

Analogously, $f\left(\left[x_{encl}^{(\kappa)}\left([0\ ;\ T]\right)\right],[0\ ;\ T]\right)$ is a guaranteed enclosure of the time derivative of the right hand side of (16). Therefore, differentiation with respect to time on both sides of (16) and solving for $\left[\dot{R}^{(\kappa+1)}\right]$ leads directly to the iteration formula (11). Finally, the evaluation of the sum of the approximate solution $x_{app}\left(t\right)$ and the bounds of the approximation error using outward rounding of the resulting interval provides a verified state enclosure of the solution of the initial value problem. ■

To prevent the growth of interval diameters of the state enclosures over simulation time for asymptotically stable dynamical systems, the following exponential enclosure approach has been introduced.

Theorem 2. *Consider again the initial value problem defined in Theorem 1 for the nonlinear set of ODEs $\dot{x}\left(t\right)=f\left(x\left(t\right),t\right)$ with $f:D\mapsto\mathbb{R}^{n_x}$, $D\subset\mathbb{R}^{n_x}\times\mathbb{R}^1$, $f\in C^1(D,\mathbb{R}^{n_x})$, where the initial states are again characterized by $x\left(0\right)\in[x_0]$. At the point of time t, $0\leq t\leq T$, $0\leq T\leq t_f$, a guaranteed enclosure of the set of all reachable states is described by the exponential term*

$$[x_{encl}\left(t\right)]:=e^{[\Lambda]\cdot t}\cdot[x_{encl}\left(0\right)] \tag{17}$$

with the diagonal matrix

$$[\Lambda]=\operatorname{diag}\left\{[\lambda_i]\right\}\ ,\quad i=1,\ldots,n_x. \tag{18}$$

The guaranteed state enclosure $[x_{encl}\left(t\right)]$ is obtained with the help of the definition (17) by the evaluation of the iteration formula

$$\left[\lambda_i^{(\kappa+1)}\right]:=\frac{f_i\left(e^{\left(\left[\Lambda^{(\kappa)}\right]\cdot[0\ ;\ T]\right)}\cdot[x_{encl}\left(0\right)],[0\ ;\ T]\right)}{e^{\left(\left[\lambda_i^{(\kappa)}\right]\cdot[0\ ;\ T]\right)}\cdot[x_{encl,i}\left(0\right)]} \tag{19}$$

for all $i=1,\ldots,n_x$.

Using this iteration formula, the entries $[\lambda_i]$ of the interval matrix $[\Lambda]$ can be determined, if the value zero does not belong to the set of reachable states for any x_i, $i=1,\ldots,n_x$, in the time interval $0\leq t\leq T$.

Proof. The proof of Theorem 2 proceeds in a similar way as the proof of Theorem 1 describing the basic iteration formula of VALENCIA-IVP. After a substitution of the exponential state enclosure $[x_{encl}\left(t\right)]$ defined in (17) for the bounding box $[B]$ on both sides of the Picard iteration (13), the expression

$$\begin{aligned} &e^{\left(\left[\Lambda^{(\kappa+1)}\right]\cdot[0\ ;\ T]\right)}\cdot[x_{encl}\left(0\right)] \\ &\quad=[x_0]+[0\ ;\ T]\cdot f\left(e^{\left(\left[\Lambda^{(\kappa)}\right]\cdot[0\ ;\ T]\right)}\cdot[x_{encl}\left(0\right)],[0\ ;\ T]\right) \end{aligned} \tag{20}$$

is obtained.

Differentiation with respect to time on both sides of (20) (see Theorem 1 and its proof for the prerequisites for the admissibility of this differentiation) leads to

$$\begin{aligned} &\operatorname{diag}\left\{\left[\lambda_i^{(\kappa+1)}\right]\right\}\cdot e^{\left(\left[\Lambda^{(\kappa+1)}\right]\cdot[0\ ;\ T]\right)}\cdot[x_{encl}\left(0\right)] \\ &\quad=f\left(e^{\left(\left[\Lambda^{(\kappa)}\right]\cdot[0\ ;\ T]\right)}\cdot[x_{encl}\left(0\right)],[0\ ;\ T]\right). \end{aligned} \tag{21}$$

If the iteration formula (19) converges, the property

$$\left[\lambda_i^{(\kappa+1)}\right]\subseteq\left[\lambda_i^{(\kappa)}\right] \tag{22}$$

holds for all $i=1,\ldots,n_x$ and, thus, also

$$\left[\Lambda^{(\kappa+1)}\right]\subseteq\left[\Lambda^{(\kappa)}\right] \tag{23}$$

holds.

Due to the monotonicity of the exponential function and due to the property of inclusion monotonicity in interval evaluation of analytic functions, the relation

$$e^{\left(\left[\Lambda^{(\kappa+1)}\right]\cdot[0\ ;\ T]\right)}\subseteq e^{\left(\left[\Lambda^{(\kappa)}\right]\cdot[0\ ;\ T]\right)} \tag{24}$$

is obtained. Substituting this relation for the exponential term on the left hand side of (21), the definition

$$\begin{aligned} &\operatorname{diag}\left\{\left[\tilde{\lambda}_i^{(\kappa+1)}\right]\right\}\cdot\underbrace{e^{\left(\left[\Lambda^{(\kappa)}\right]\cdot[0\ ;\ T]\right)}\cdot[x_{encl}\left(0\right)]}_{\left[x_{encl}^{(\kappa)}\left([0\ ;\ T]\right)\right]} \\ &\quad:=f\left(e^{\left(\left[\Lambda^{(\kappa)}\right]\cdot[0\ ;\ T]\right)}\cdot[x_{encl}\left(0\right)],[0\ ;\ T]\right) \end{aligned} \tag{25}$$

for the intervals $\left[\tilde{\lambda}_i^{(\kappa+1)}\right]$ is obtained. Due to inclusion monotonicity of the exponential function in (24), it is guaranteed that the intervals $\left[\tilde{\lambda}_i^{(\kappa+1)}\right]$ represent verified enclosures of the intervals $\left[\lambda_i^{(\kappa+1)}\right]$ to be determined.

A division of (25) by the interval $\left[x_{encl,i}^{(\kappa)}\left([0\ ;\ T]\right)\right]$ with component-wise notation results in the iteration formula (19). In this iteration formula, the intervals $\left[\tilde{\lambda}_i^{(\kappa+1)}\right]$ are used instead of $\left[\lambda_i^{(\kappa+1)}\right]$. The division on the right hand side of (19) and, therefore, also the application of the exponential enclosure technique are only admissible if the value zero is not contained in the set of reachable states for any component of the state vector in the complete time interval $[0\ ;\ T]$. This condition has to be checked using the iteration in Theorem 1. ■

The quality of the interval enclosures is influenced by the following factors:

1. *The initial approximation $x_{app}(t)$ in the complete time interval.* Smaller deviations between the unknown exact solution and its initial approximation lead to tighter enclosures of the solution over a longer time span with less computational effort.

2. *Time span.* Tighter bounds can be obtained by subdividing the time span $[0\ ;\ T]$. This improves the convergence of the iteration formula (11) and additionally leads to smaller interval bounds; see also *Step 3* in the following description of the algorithm.

3. *Preconditioning of the state equations.* The efficiency of the iteration formulas (11) and (19) can be improved if the wrapping effect in verified simulation of ODEs is reduced by suitable preconditioning of the state equations.

Algorithm. In the following, the key components of VALENCIA-IVP for ODEs are summarized. More details are given in (Auer *et al.*, 2008).

Step 1. Calculation of reference solutions. VALENCIA-IVP can make use of both analytical and numerical approximations $x_{app}(t)$ for initial value problems of ODEs. In this paper, we only discuss the use of numerical approximations since they enable the user to apply VALENCIA-IVP to a wider class of systems without the necessity for system-specific modifications.

Consider a non-verified numerical approximation $\{x_i^N\}$, $i = 0, \ldots, L$, for the original initial value problem with the point interval $x_0^N = \mathrm{mid}([x_0])$ as the initial condition. This numerical approximation is computed over a grid $\{t_i\}$ with $t_L = T$ by an arbitrary non-verified IVP solvers. In the iteration formula (11), an analytic expression is required for $x_{app}(t)$ and its time derivative $\dot{x}_{app}(t)$. This expression is determined by minimizing a distance measure between the numerically determined points $\{x_i^N\}$ and the approximate solution $x_{app}(t)$ of the solution of the initial value problem.

For this purpose, VALENCIA-IVP currently uses a linear interpolation between the grid points according to

$$x_{app}(t) = x_i^N + \frac{x_{i+1}^N - x_i^N}{t_{i+1} - t_i} \cdot (t - t_i) \tag{26}$$

with

$$\dot{x}_{app}(t) = \frac{x_{i+1}^N - x_i^N}{t_{i+1} - t_i} \tag{27}$$

and $t \in [t_i\ ;\ t_{i+1}]$, $i = 0, \ldots, L-1$. The advantage of this method is that $x_{app}(t)$ is obtained with small computational effort. The step sizes $t_{i+1} - t_i$, $i = 0, \ldots, L-1$, are determined by the non-verified solver.

Analogously, higher-order interpolations between the numerically calculated grid points expressed by parameterizable functions $x_{app}(t)$ can be included in the source code of this solver instead of linear interpolation. On the one hand, the deviation between the approximate and exact solutions of IVP is reduced by these improved approximations. On the other hand, the dependency on time of these higher-order interpolations is always nonlinear. Since the iteration formula (11), which is based on the nonlinear state equations, has to be evaluated for time intervals and not only for infinitesimally short points of time in the following Steps 2–4, the influence of overestimation is growing, if such interpolations are used. Due to these two effects, a compromise has to be found between the improvement of the initial approximation and the computational effort, which is necessary to reduce the arising overestimation. For techniques aiming at the reduction of overestimation, see the discussion of advanced interval methods in Step 4.

Step 2. Initialization of the iteration scheme. To start the iteration (11), initial interval approximations for $[R(t)]$ and $\left[\dot{R}(t)\right]$ are required. Afterwards, in the first iteration step $\kappa = 0$, $\left[\dot{R}^{(1)}(t)\right]$ is calculated. The iteration is continued, if $\left[\dot{R}^{(1)}(t)\right] \subseteq \left[\dot{R}^{(0)}(t)\right]$. Otherwise, the initial guesses for $[R(t)]$ and $\left[\dot{R}(t)\right]$ have to be modified. Note that the interval enclosure $[R(0)]$ for the initial point of time has to be chosen such that all possible initial states are included, i.e., $[x_0] \subseteq x_{app}(0) + [R(0)]$.

Step 3. Subdivision of the time span into several time intervals. If the time span $[0\ ;\ T]$ is split into several shorter time intervals to improve the convergence of the iteration and to reduce the width of the error bounds, again verified integration of $\left[\dot{R}^{(\kappa+1)}\right]$ is necessary to obtain a guaranteed enclosure for the error term. As follows directly from (12), the integration with respect to time is performed by

$$\begin{aligned}
&\left[R^{(\kappa+1)}(t_{i+1})\right] \\
&= \left[R^{(\kappa+1)}(0)\right] \\
&\quad + \sum_{j=0}^{i} (t_{j+1} - t_j) \cdot r\left(\left[R^{(\kappa)}([\Delta t_j])\right], [\Delta t_j]\right)
\end{aligned} \tag{28}$$

with

$$[\Delta t_j] := [t_j\ ;\ t_{j+1}] \tag{29}$$

for all $\{t_i\}$, $i = 0, \ldots, L-1$. For numerical reference solutions with the linear interpolation (26) between the grid points, $\{t_i\}$ is determined by the non-verified ODE solver which has been applied in Step 1. Note that the grid on the time axis does not have to be equally spaced.

Step 4. Calculation of the state enclosures. The width of the state enclosure $[x(t)] \subseteq x_{app}(t) + [R(t)]$ determines whether improved initial approximations in Step 1 and smaller time intervals $[\Delta t_j]$ in Step 3 are necessary to reduce overestimation in the interval enclosures.

In the evaluation of (11) and (19), overestimation results from multiple occurrence of identical interval variables. This overestimation is reduced by mean-value rule evaluation of the right hand side of the iteration formulas (11) and (19) as well as iterative improvement of the range of the expression on the right hand side including monotonicity tests. In VALENCIA-IVP, all partial derivatives required for these interval techniques are determined by automatic/algorithmic differentiation using FADBAD++ (Bendsten and Stauning, 2007).

(a) Mean-value rule evaluation of iteration formula. Since natural interval evaluation of nonlinear expressions often leads to overestimation, the iteration formula (11) is evaluated using the mean-value rule

$$r(z) \in r(z_m) + \left.\frac{\partial r}{\partial z}\right|_{z=[z]} \cdot ([z] - z_m) \qquad (30)$$

for all $z \in [z]$ with the vector

$$[z] = \begin{bmatrix} [R(t_i)] \\ [t_i \; ; \; t_{i+1}] \end{bmatrix} \quad \text{and} \quad z_m = \operatorname{mid}([z]) \qquad (31)$$

containing all interval arguments of the right hand side of (11). To obtain the tightest possible enclosures, we use the intersection of the results of both natural interval extension and mean-value rule evaluation in all further computations.

(b) Monotonicity test. Additionally, VALENCIA-IVP performs a monotonicity test for further reduction of overestimation. In the case of the monotonicity of the component r_i, $i = 1, \ldots, n_x$, with respect to at least one z_j, $j = 1, \ldots, n_x + 1$, i.e., if the lower bound of the interval evaluation of $\partial r_i / \partial z_j$ is strictly positive or if its upper bound is strictly negative, the interval $[z_j]$ can be replaced by one of the interval bounds as summarized in Table 1. For example, if $\inf(\partial r_i/\partial z_j) > 0$, $[z_j]$ can be replaced by $\underline{z}_j$ to compute the infimum of the range of r_i over $[z]$ and by $\overline{z}_j$ to compute its supremum. The range of r_i is then given by the interval hull of the results of both rows in Table 1.

Table 1. Replacement of arguments in the iteration formula of VALENCIA-IVP in the case of monotonicity.

	$\inf\left(\frac{\partial r_i}{\partial z_j}\right) > 0$	$\sup\left(\frac{\partial r_i}{\partial z_j}\right) < 0$
$\inf\left\{ r_i(z)\Big\vert_{z_j=\xi_j} \right\}$	$\xi_j = \underline{z}_j$	$\xi_j = \overline{z}_j$
$\sup\left\{ r_i(z)\Big\vert_{z_j=\xi_j} \right\}$	$\xi_j = \overline{z}_j$	$\xi_j = \underline{z}_j$

(c) Iterative calculation of the range. If the monotonicity test is not successful in at least one argument of r_i, all arguments of r_i with interval diameters which are significantly larger than zero can be split into several subintervals for which mean-value rule evaluation and monotonicity tests are applied again. Since only tight upper and lower bounds of r_i are desired, the splitting procedure is continued with the input intervals which lead to the smallest infimum/largest supremum to improve the lower/upper bounds of r_i. Splitting in VALENCIA-IVP is continued until a user-defined number of subintervals is reached or until r_i is monotonic for all input arguments. Finally, the *union* of all subintervals for r_i is determined to compute the improved enclosure of its range. For numerous practically relevant dynamical systems, a small number of splittings is required to obtain good enclosures of the range if monotonicity is checked for each subinterval. Thus, compared to methods employing derivatives of high orders, the advantages of this procedure are simplicity of implementation and often a smaller computational effort.

Analogously, the techniques (a)–(c) are applied to the exponential enclosure technique (Theorem 2) in which the intervals $\left[\lambda_i^{(\kappa+1)}\right]$ are computed iteratively.

In (Rauh *et al.*, 2007a), the authors have demonstrated the applicability of VALENCIA-IVP to the simulation of dynamical systems with both uncertain parameters and uncertain initial states. In further extensions of VALENCIA-IVP, consistency tests based on backward integration of subintervals of the state enclosure $[x_{encl}(t)]$ have been implemented. These consistency tests aim at the reduction of overestimation by the detection and elimination of subintervals of $[x_{encl}(t)]$ which certainly result from overestimation (Rauh *et al.*, 2007a; Rauh *et al.*, 2007b). Finally, physical conservation properties can be exploited as further constraints to detect and reduce overestimation (Rauh *et al.*, 2008).

4. Application of interval Newton techniques to ensure the consistency of the state variables of DAEs

Interval Newton methods are fundamental techniques for the solution of sets of DAEs using VALENCIA-IVP. In this section, the computation of guaranteed interval enclosures of steady states—under the assumption of time-invariant uncertain parameters and constant control inputs—as well as approaches for the verification of the consistency of initial states are demonstrated. These approaches are extended to the verification of the dynamics of sets of DAEs in Section 5.

4.1. Computation of guaranteed enclosures of steady states.

To compute guaranteed interval enclosures of the sets of steady states, the state equations (7a) and (7b) have to be evaluated for $\dot{x} = 0$. Then, interval Newton techniques, such as the Krawczyk iteration (Krawczyk, 1969), see also the Appendix, can be applied to search for all combinations of x and y in a predefined region which fulfill the resulting nonlinear algebraic equations.

Example 1. To demonstrate this property, the pendulum example from (Nedialkov, 2007) rewritten as a set of first order differential equations

$$\begin{aligned} \dot{x}_1 &= x_3, \\ \dot{x}_2 &= x_4, \\ \dot{x}_3 &= -x_1 y, \\ \dot{x}_4 &= -x_2 y + 1, \\ g(x) &= x_1^2 + x_2^2 - 1 = 0 \end{aligned} \tag{32}$$

is considered. Assuming the interval boxes

$$[x] = \begin{bmatrix} [-5\ ;\ 5] \\ [-5\ ;\ 5] \\ [-5\ ;\ 5] \\ [-5\ ;\ 5] \end{bmatrix} \quad \text{and} \quad [y] = [-5\ ;\ 5] \tag{33}$$

as the *a priori* given search space, enclosures of both consistent steady states $x^{\langle 1\rangle}$, $y^{\langle 1\rangle}$ and $x^{\langle 2\rangle}$, $y^{\langle 2\rangle}$ are computed. These steady states (which can easily be determined analytically) are given by

$$x^{\langle 1\rangle} = \begin{bmatrix} 0 & -1 & 0 & 0 \end{bmatrix}^T, \quad y^{\langle 1\rangle} = -1$$

and

$$x^{\langle 2\rangle} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^T, \quad y^{\langle 2\rangle} = 1.$$

The corresponding interval enclosures are obtained using the Krawczyk iteration with diameters smaller than 10^{-40}. ♦

4.2. Consistency test for the initial conditions of DAEs.

Usually, the information provided by the algebraic constraints (7b) is not sufficient to verify the consistency of initial states of a set of DAEs. In this case, *additional* hidden constraints are necessary to restrict the set of feasible solutions and to verify the consistency of $x(t_0)$ and $y(t_0)$. For that purpose, those constraints $g_i(x)$ are considered, which do not depend explicitly on y. Differentiation with respect to time leads to

$$\begin{aligned} \frac{\mathrm{d}^j g_i(x)}{\mathrm{d}t^j} &= \left(\frac{\partial L_f^{j-1} g_i(x)}{\partial x}\right)^T \cdot f(x, y) \\ &= L_f^j g_i(x) = 0 \end{aligned} \tag{34}$$

with

$$L_f^0 g_i(x) = g_i(x). \tag{35}$$

The Lie derivatives $L_f^j g_i(x)$ are computed by FADBAD++ providing automatic differentiation and automatic calculation of Taylor coefficients. The differentiation is continued up to the smallest order $j > 0$ for which $L_f^j g_i(x)$ depends on at least one component of y (i.e., up to the differentiation index of DAEs). The computation of these Lie derivatives is based on a procedure published in (Röbenack, 2002). To our knowledge, the author of (Röbenack, 2002) has not used this approach in an interval arithmetic framework to account for uncertainties in ODEs and DAEs.

Example 2. To illustrate the use of the hidden constraints (34), the above-mentioned example (32) is considered again. For this system, symbolic evaluation of (34) which is avoided in VALENCIA-IVP due to the use of automatic differentiation gives the constraints

$$\begin{aligned} \frac{\mathrm{d}g(x)}{\mathrm{d}t} &= 2(x_1\dot{x}_1 + x_2\dot{x}_2) \\ &= 2(x_1 x_3 + x_2 x_4) \overset{!}{=} 0 \end{aligned} \tag{36}$$

and

$$\begin{aligned} \frac{\mathrm{d}^2 g(x)}{\mathrm{d}t^2} &= 2(\dot{x}_1 x_3 + x_1\dot{x}_3 + \dot{x}_2 x_4 + x_2\dot{x}_4) \\ &= 2\left(x_3^2 - x_1^2 y + x_4^2 - x_2^2 y + x_2\right) \overset{!}{=} 0 \end{aligned} \tag{37}$$

corresponding to the first and second derivatives of g with respect to time.

These expressions can be used to verify the results obtained by automatic differentiation which show that, for example, $x(0) = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^T$ and $y(0) = 1$ represent consistent initial values, while certainly no consistent initial conditions are included in $x(0) \in \big[[-0.5\ ;\ 0.5]\ \ [-0.5\ ;\ 0.5]\ \ [-0.5\ ;\ 0.5]\ \ [-1.0\ ;\ 1.0]\big]^T$ for $y(0) = 1$. ♦

4.3. Structural analysis of DAEs for automatic computation of consistent initial states.

An extension of the consistency test for initial states is automatic modification of these enclosures to ensure consistency. For that purpose, the user of VALENCIA-IVP can specify which components of the state vectors x and y are *fixed* and therefore not allowed to be modified automatically and which components are *free* and allowed to be changed

Table 2. Interval enclosures of possible consistent initial states.

	$[x_1]$	$[x_2]$	$[x_3]$
candidate 1	$[\ \ 0.7716\ ;\ \ \ 0.8016]$	$[-0.6362\ ;\ -0.5979]$	$[\ \ 0.7535\ ;\ \ \ 0.8162]$
candidate 2	$[-1.0010\ ;\ -0.9995]$	$[-0.0312\ ;\ \ \ 0.0291]$	$[-0.0315\ ;\ \ \ 0.0294]$
candidate 3	$[-0.8016\ ;\ -0.7708]$	$[-0.6361\ ;\ -0.5979]$	$[-0.8162\ ;\ -0.7535]$
candidate 4	$[\ \ 0.9995\ ;\ \ \ 1.0009]$	$[-0.0312\ ;\ \ \ 0.0291]$	$[-0.0294\ ;\ \ \ 0.0315]$

Results of the Krawczyk iteration up to four decimal digits; lower interval bounds are rounded towards minus infinity, upper bounds towards plus infinity.

within user-defined interval bounds. To solve this problem, the Krawczyk iteration, which is applied to the algebraic constraints and their Lie derivatives, is modified such that the *fixed* components are treated as constant interval parameters, while the solver only determines new enclosures for the free variables.

Example 3. Consider again the pendulum example with the initial state enclosures

$$[x] = \begin{bmatrix} [-5.0\ ;\ 5.0] \\ [-5.0\ ;\ 5.0] \\ [-5.0\ ;\ 5.0] \\ [0.99\ ;\ 1.01] \end{bmatrix} \text{ and } [y] = [0.99\ ;\ 1.01] \tag{38}$$

for which the components x_4 and y have been specified to be *fixed*. All four interval enclosures in which consistent initial states for x_1, x_2, and x_3 are contained can be computed using VALENCIA-IVP. The corresponding enclosures $[x_1]$, $[x_2]$, and $[x_3]$ are listed in Table 2. ♦

5. VALENCIA-IVP for the calculation of guaranteed state enclosures for DAEs

For the computation of interval enclosures of the trajectories of the states $x\,(t)$ and $y\,(t)$ of DAEs, the semi-explicit representation

$$0 = -\dot{x}(t) + f\big(x(t), y(t), t\big), \tag{39a}$$

$$0 = g\big(x(t), y(t), t\big) \tag{39b}$$

is considered. According to (10), interval enclosures

$$[x_{encl}(t)] := x_{app}(t) + [R_x(t)], \tag{40a}$$

$$[y_{encl}(t)] := y_{app}(t) + [R_y(t)] \tag{40b}$$

can be defined which distinguish between the differential variables $[x_{encl}(t)]$ and the algebraic variables $[y_{encl}(t)]$, respectively. Analogously to VALENCIA-IVP for ODEs, the enclosure of the solution consists of approximate solutions $x_{app}\,(t)$ and $y_{app}\,(t)$ with interval enclosures of the error terms $[R_x(t)]$ and $[R_y(t)]$. By substituting the interval enclosures (40) for the state vectors $x\,(t)$ and $y\,(t)$ in (39), a nonlinear set of algebraic equations

$$\begin{aligned} 0 &= -\,\dot{x}_{app}(t) - \Big[\dot{R}_x(t)\Big] \\ &\quad + f\Big(x_{app}(t) + [R_x(t)],\ y_{app}(t) + [R_y(t)],\ t\Big), \\ 0 &= g\Big(x_{app}(t) + [R_x(t)],\ y_{app}(t) + [R_y(t)],\ t\Big) \end{aligned} \tag{41}$$

can be derived for $[R_x(t)]$, $[R_y(t)]$, and $[\dot{R}_x(t)]$, $0 \le t \le T$. As for sets of ODEs, the error bounds are defined by

$$\begin{aligned} [R_x(t)] &:= [R_x(0)] + t \cdot \Big[\dot{R}_x([0;t])\Big]\,, \\ [R_y(t)] &:= [R_y(0)] + t \cdot \Big[\dot{R}_y([0;t])\Big]\,. \end{aligned} \tag{42}$$

Considering the time interval $[T] := [0;T]$, the nonlinear algebraic equations

$$\begin{aligned} 0 \;=\; & -\,\dot{x}_{app}([T]) - \Big[\dot{R}_x([T])\Big] \\ & + f\Big(x_{app}([T]) + [R_x(0)] \;+\; [T]\cdot\Big[\dot{R}_x([T])\Big]\,, \\ & \qquad y_{app}([T]) + [R_y(0)] \;+\; [T]\cdot\Big[\dot{R}_y([T])\Big]\,,\ [T]\Big), \\ 0 \;=\; & g\Big(x_{app}([T]) + [R_x(0)] \;+\; [T]\cdot\Big[\dot{R}_x([T])\Big]\,, \\ & \qquad y_{app}([T]) + [R_y(0)] \;+\; [T]\cdot\Big[\dot{R}_y([T])\Big]\,,\ [T]\Big) \end{aligned} \tag{43}$$

have to be solved for $[\dot{R}_x([T])]$ and $[\dot{R}_y([T])]$ using a suitable verified approach. As in the previous section, interval Newton techniques are applied for that purpose.

6. Solving DAEs with VALENCIA-IVP

In Section 5, the basic approach of VALENCIA-IVP has been extended to handle sets of DAEs. Using the equations (43), an interval enclosure for an initial value problem of a set of DAEs can be computed if suitable approximate solutions $x_{app}(t)$ and $y_{app}(t)$ are available which are determined using DASSL or DAETS. For the initial point of time $t_0 = 0$, the error term is defined by

$$[R(0)] = [z_{encl}(0)] - z_{app}(0) \tag{44}$$

with

$$[z_{encl}(0)] := \left[[x_{encl}(0)]^T \quad [y_{encl}(0)]^T\right]^T \tag{45}$$

and

$$z_{app}(0) := \left[x_{app}(0)^T \quad y_{app}(0)^T\right]^T . \tag{46}$$

Now the set of equations (43) has to be solved in a verified way. This is possible using interval Newton techniques like the Krawczyk iteration (Krawczyk, 1969), see the Appendix. However, solving for $[\dot{R}_y([T])]$ is not possible directly since, as shown below, the equations (43) are structurally singular for $t = 0$.

Hence, the Krawczyk iteration is applied to determine the zeros of the expressions

$$h_x\left([\dot{R}_x([T])], [R_y([T])]\right) \tag{47a}$$

$$:= -\dot{x}_{app}([T]) - [\dot{R}_x([T])] \ + \ f\Big([\tilde{x}], [\tilde{y}], [T]\Big),$$

$$h_y\left([\dot{R}_x([T])], [R_y([T])]\right) \ := \ g\Big([\tilde{x}], [\tilde{y}], [T]\Big) \tag{47b}$$

with

$$[\tilde{x}] := \Big(x_{app}([T]) + [R_x(0)] \ + \ [T] \cdot \Big[\dot{R}_x([T])\Big]\Big), \tag{48a}$$

$$[\tilde{y}] := \Big(y_{app}([T]) + [R_y([T])]\Big) \tag{48b}$$

to compute the interval enclosures $[\dot{R}_x([T])]$ and $[R_y([T])]$ with $\dot{R}_x \in \mathbb{R}^{n_x}$ and $R_y \in \mathbb{R}^{n_y}$.

If the time derivative $[\dot{R}_y(t)]$ was used as an argument of the functions h_x and h_y, the partial derivatives

$$\left.\frac{\partial h_x\left(\dot{R}(t)\right)}{\partial \dot{R}(t)}\right|_{\dot{R}(t)\in\left[\dot{R}([T])\right]} = -I_{(n_x,n_x+n_y)} + \frac{\partial f\Big([\tilde{z}], [T]\Big)}{\partial [\tilde{z}]} \cdot \underbrace{\frac{\partial [\tilde{z}]}{\partial [\dot{R}([T])]}}_{=[T]\cdot I_{(n,n)}}, \tag{49a}$$

$$\left.\frac{\partial h_y\left(\dot{R}(t)\right)}{\partial \dot{R}(t)}\right|_{\dot{R}(t)\in\left[\dot{R}([T])\right]} = \frac{\partial g\Big([\tilde{z}], [T]\Big)}{\partial [\tilde{z}]} \cdot \underbrace{\frac{\partial [\tilde{z}]}{\partial [\dot{R}([T])]}}_{=[T]\cdot I_{(n,n)}} \tag{49b}$$

with the interval vectors

$$[R([T])] := \begin{bmatrix} [R_x([T])] \\ [R_y([T])] \end{bmatrix} \qquad \text{and} \qquad [\tilde{z}] := \begin{bmatrix} [\tilde{x}] \\ [\tilde{y}] \end{bmatrix}$$

are structurally singular at least for the initial point of time in the interval $[T] := [0 \ ; \ T]$. In (49a) and (49b), the identity matrix $I_{(p,q)} \in \mathbb{R}^{p\times q}$ is defined according to

$$I_{i,j} = \begin{cases} 1 & \text{for } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{50}$$

Using (47), the algebraic equations are not structurally singular as shown by the inner derivative

$$\frac{\partial [\tilde{z}]}{\partial \begin{bmatrix} [\dot{R}_x([T])] \\ [R_y([T])] \end{bmatrix}} = \begin{bmatrix} [T]\cdot I_{(n_x,n_x)} & O_{(n_x,n_y)} \\ O_{(n_y,n_x)} & I_{(n_y,n_y)} \end{bmatrix} \tag{51}$$

with

$$[\tilde{z}] := \begin{bmatrix} x_{app}([T]) + [R_x(0)] \ + \ [T] \cdot \Big[\dot{R}_x([T])\Big] \\ y_{app}([T]) + [R_y([T])] \end{bmatrix} . \tag{52}$$

This new definition avoids structural singularity of the derivative

$$\frac{\partial h\left([\dot{R}_x([T])], [R_y([T])]\right)}{\partial \begin{bmatrix} [\dot{R}_x([T])] \\ [R_y([T])] \end{bmatrix}}, \quad h\left(\cdot\right) := \begin{bmatrix} h_x\left(\cdot\right) \\ h_y\left(\cdot\right) \end{bmatrix} \tag{53}$$

of (47). The zero matrix $O_{(p,q)} \in \mathbb{R}^{p\times q}$ is of dimension $p \times q$.

However, the information provided by these equations is not always sufficient to solve for $[\dot{R}_x\left([T]\right)]$ and $[R_y\left([T]\right)]$. Considering the pendulum example in the previous section, the equations are still singular for small step sizes $T \rightarrow 0$. This singularity is related to differentiation indices of DAEs which are greater than one. In this case, the Lie derivatives of the algebraic constraints which have been introduced to determine sets of feasible initial conditions have to be considered to ensure the solvability of the problem.

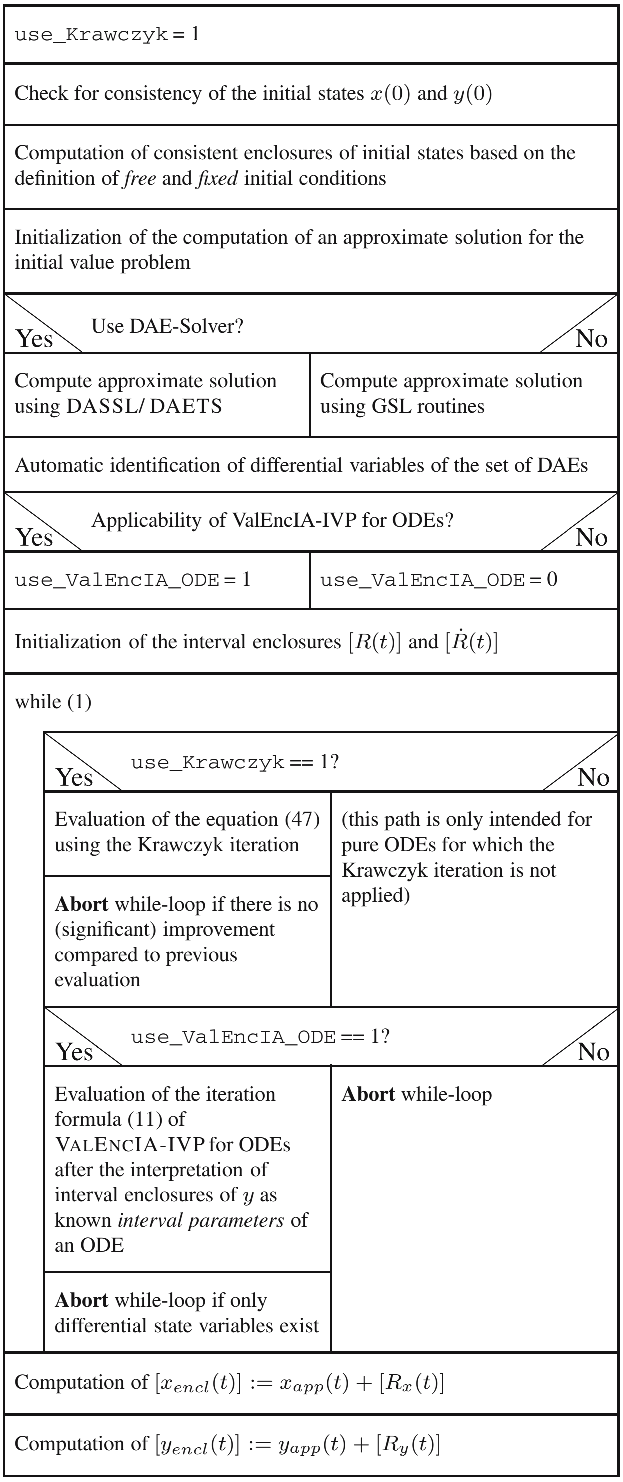
To deal with problems related to higher differential indices of DAEs, a suitable extension of VALENCIA-IVP is to treat the intervals $[R_y\left([T]\right)]$ as *constant interval parameters* in a first stage and to solve the non-algebraic equations for $[\dot{R}_x\left([T]\right)]$. In a second stage, the consistency of the solution has to be proven with the help of the algebraic equations and their time derivatives by showing that feasible solutions are guaranteed to be contained in the interior of $[R_y\left([T]\right)]$.

7. Implementation

VALENCIA-IVP for sets of DAEs is implemented as a C++ program. In Fig. 1, the basic structural diagram of this program is shown.

As pointed out in Section 2, the extension of VALENCIA-IVP has only been discussed for semi-explicit DAEs. This is due to the fact that the computation of time derivatives of pure algebraic constraints is implemented for this special case. However, the Krawczyk

VALENCIA-IVP — Algorithm for solving sets of DAEs



```
use_Krawczyk = 1
Check for consistency of the initial states $x(0)$ and $y(0)$
Computation of consistent enclosures of initial states based on the definition of free and fixed initial conditions
Initialization of the computation of an approximate solution for the initial value problem
Use DAE-Solver?
  Yes: Compute approximate solution using DASSL/ DAETS
  No:  Compute approximate solution using GSL routines
Automatic identification of differential variables of the set of DAEs
Applicability of ValEncIA-IVP for ODEs?
  Yes: use_ValEncIA_ODE = 1
  No:  use_ValEncIA_ODE = 0
Initialization of the interval enclosures $[R(t)]$ and $[\dot{R}(t)]$
while (1)
    use_Krawczyk == 1?
      Yes: Evaluation of the equation (47) using the Krawczyk iteration
           Abort while-loop if there is no (significant) improvement compared to previous evaluation
      No:  (this path is only intended for pure ODEs for which the Krawczyk iteration is not applied)
    use_ValEncIA_ODE == 1?
      Yes: Evaluation of the iteration formula (11) of VALENCIA-IVP for ODEs after the interpretation of interval enclosures of $y$ as known interval parameters of an ODE
           Abort while-loop if only differential state variables exist
      No:  Abort while-loop
Computation of $[x_{encl}(t)] := x_{app}(t) + [R_x(t)]$
Computation of $[y_{encl}(t)] := y_{app}(t) + [R_y(t)]$
```

Fig. 1. Structural diagram of VALENCIA-IVP for DAEs.

iteration can also deal with the case of general, fully implicit DAE systems, as long as no derivatives of pure algebraic constraints are required to solve for $[\dot{R}_x([T])]$ and $[R_y([T])]$, and if consistent enclosures of the sets of initial states are known.

Therefore, general DAEs as well as implicit ODEs

$$0 = g\left(x\left(t\right), \dot{x}\left(t\right), y\left(t\right)\right) \tag{54}$$

are currently transformed into sets of semi-explicit DAEs according to

$$\begin{aligned} \dot{x}\left(t\right) &= \tilde{x}\left(t\right), \\ 0 &= g\left(x\left(t\right), \tilde{x}\left(t\right), y\left(t\right)\right) \end{aligned} \tag{55}$$

by introducing further algebraic state variables $\tilde{x}\left(t\right)$.

Since ODEs are a special case of sets of DAEs, the implementation of VALENCIA-IVP shown in Fig. 1 is capable to solve both types of dynamical system models. Therefore, the Krawczyk iteration can be used in addition to the iteration formulas (11) and (19) of VALENCIA-IVP to determine state enclosures for sets of ODEs. The Krawczyk iteration is enabled by setting the variable `use_Krawczyk` true.

For DAEs, the routines for the consistency test and the modification of initial state enclosures as described in Section 4 are available. Non-verified approximate solutions are determined numerically with the help of DASSL or DAETS for sets of DAEs. Routines provided by the GSL library are available for the computation of approximate solutions for sets of ODEs.

After these initializations, guaranteed state enclosures are determined as described in the previous section. Basic interval arithmetic operations and functionalities for automatic differentiation are provided by PROFIL/BIAS (Keil, 2007) and FADBAD++ (Bendsten and Stauning, 2007), respectively.

If the computation of guaranteed state enclosures for DAEs using the Krawczyk iteration is successful, the enclosures for the differential state variables $x\left(t\right)$ can often be refined using the iteration formulas (11) and (19) of VALENCIA-IVP for ODEs. For that purpose, the state variables $y\left(t\right)$ are replaced by their interval enclosures in each time step. These enclosures—which are formally treated as parameters with constant interval bounds—have been computed using the Krawczyk iteration. The refinement of state enclosures of $x\left(t\right)$ is activated by `use_ValEncIA_ODE` = 1. It can be implemented in a straightforward way for semi-explicit DAEs.

In general, also the case when the differential part of a DAE system is given by

$$M\left(y\left(t\right)\right) \cdot \dot{x}\left(t\right) = f\left(x\left(t\right), y\left(t\right), t\right) \tag{56}$$

can be handled instead of

$$\dot{x}\left(t\right) = f\left(x\left(t\right), y\left(t\right), t\right) \tag{57}$$

if substituting the corresponding interval enclosures for $y(t)$ leads to an interval matrix $[M]$ with full rank. Further structures, e.g., $M(x(t), y(t))$ or general nonlinear terms depending also on $\dot{x}(t)$ have to be dealt with using (54) and (55). A possible extension is to use interval Newton techniques to solve for $\dot{x}(t)$. Then, the iteration formulas (11) and (19) of VALENCIA-IVP can also be used in the case of general DAEs. Furthermore, generalizations for the computation of enclosures of the sets of consistent initial conditions will be investigated in future research for general DAE systems.

8. Application

In the following application scenario, a simple electrical network is used to illustrate the use of VALENCIA-IVP for sets of DAEs. First, as shown in Fig. 2, verified simulation of the corresponding mathematical system model is performed for known interval bounds of the system's input voltage $u_{in}(t)$. In this case, the influence of variations of $u_{in}(t)$ on the output variable $u_{out}(t) = u_C(t)$ is quantified with the help of guaranteed state enclosures.

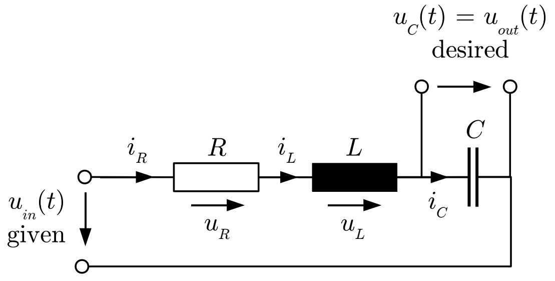


Fig. 2. Verified simulation of dynamical system models.

In contrast to this simulation of an initial value problem, Fig. 3 illustrates the computation of an appropriate input signal $u_{in}(t)$ such that $u_{out}(t) = u_C(t)$ is consistent with a predefined time-dependent function. This problem is often referred to as the solution of an *inverse control problem* (Czechowski *et al.*, 2006). In this case, piecewise linear input signals $u_{in}(t)$ are determined, i.e., $\dot{u}_{in}(t)$ is assumed to be piecewise constant.

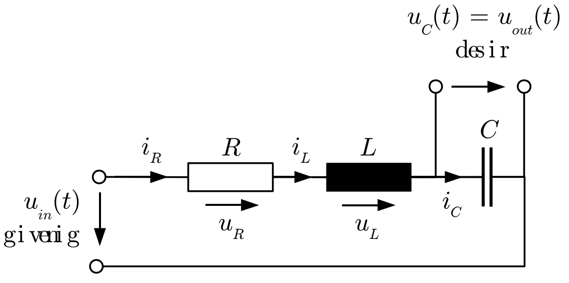


Fig. 3. Solution of inverse control problems.

In both cases, the electrical network is described by the DAEs

$$\begin{aligned}
i_C(t) &= C \cdot \frac{\mathrm{d}u_C(t)}{\mathrm{d}t}, \\
u_L(t) &= L \cdot \frac{\mathrm{d}i_L(t)}{\mathrm{d}t}, \\
u_{in}(t) &= u_R(t) + u_L(t) + u_{out}(t), \\
u_C(t) &= u_{out}(t), \\
i_R(t) &= i_L(t) = i_C(t), \\
R &= \frac{u_R(t)}{i_R(t)}
\end{aligned} \tag{58}$$

derived by Kirchhoff's voltage and current laws.

According to (7), two state vectors

$$x(t) = \begin{bmatrix} u_C(t) \\ i_L(t) \end{bmatrix}, \quad y(t) = \begin{bmatrix} i_C(t) \\ u_L(t) \\ u_R(t) \\ i_R(t) \\ u_{out}(t) \end{bmatrix} \tag{59}$$

are defined. The system parameters are assumed to be given by their normalized, nominal values $R = 1$, $L = 1$, and $C = 1$.

8.1. Verified solution of initial value problems.

For the simulation of an initial value problem for the DAEs (58), the initial states

$$x(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad y(0) = \begin{bmatrix} 0 \\ u_{in}(0) \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{60}$$

are considered. The input voltage $u_{in}(t)$ is assumed to be an unknown parameter with $u_{in}(t) \in [0.9\,;\,1.1]$. In contrast to the pendulum example discussed in Section 4, the set of DAEs (58) can be solved directly without computing any time derivative of the algebraic constraints $g(x(t), y(t)) = 0$.

As shown in Fig. 4, guaranteed interval enclosures for all state variables are obtained by directly solving the set of DAEs in VALENCIA-IVP using the Krawczyk iteration. The result of this verified simulation is compared with a grid-based reference simulation obtained by a non-verified ODE solver for which the algebraic state variables $y(t)$ have been eliminated symbolically.

This application scenario directly points out one necessary extension for further improvement of state enclosures. Each algebraic equation in a set of DAEs can be interpreted as a constraint that allows us to detect and eliminate overestimation at arbitrary points of time. For example, the constraint $i_R(t) = i_L(t)$ indicates that both state variables are identical for all points of time. Using

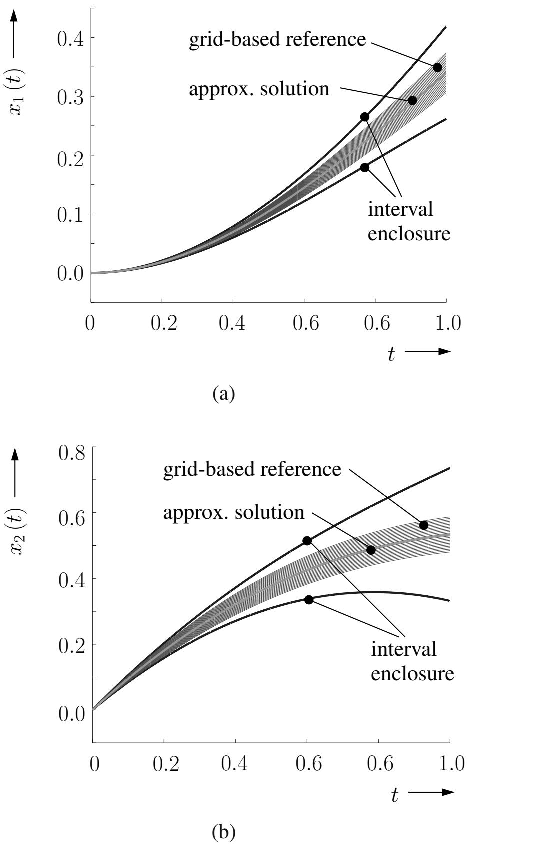


Fig. 4. State enclosures for the simulation of an initial value problem for the electrical network: (a) interval enclosure for $x_1(t)$, (b) interval enclosure for $x_2(t)$.

the Krawczyk iteration to determine $[\dot{R}_x(t)]$ and $[R_y(t)]$, axis-aligned interval boxes are obtained for the state enclosures. These boxes can be subdivided into smaller enclosures from which all parts that are inconsistent with the algebraic constraints can be eliminated, see Fig. 5 as an example. All subintervals $[\tilde{x}]$ and $[\tilde{y}]$ for which $0 \in g([\tilde{x}], [\tilde{y}])$ (the white subintervals in Fig. 5) holds can furthermore be checked for consistency by backward evaluation of the state equations. If subintervals are detected which lead to state enclosures in the backward integration which do not overlap with the results obtained in integration from $t = t_0 = 0$ to $t > t_0$, they are guaranteed to be caused by overestimation. Therefore, they are discarded.

As indicated by Fig. 5, the preconditioning of the set of state equations such that the resulting state enclosures can be represented by tighter axis-aligned interval boxes in a new coordinate system (scaling and rotating of the state enclosures) will be considered together with the above-mentioned consistency tests in future work.

8.2. Computation of control sequences.

In addition to the simulation of sets of DAEs, ValEncIA-IVP can also be used to compute interval enclosures of control sequences. In Figs. 6 and 7, two different cases are distinguished for

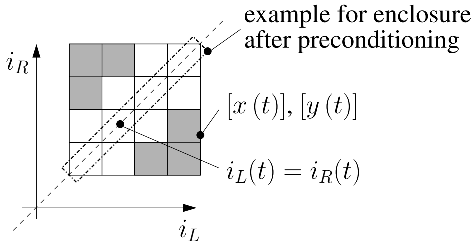


Fig. 5. Use of algebraic constraints in consistency tests for the identification and elimination of overestimation; gray: inconsistent subintervals, white: possibly consistent.

$$x(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{and} \quad y(0) = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{61}$$

with $u_{in}(0) = 1$. The desired output signal $u_{out}(t)$ which is used as a time-dependent algebraic constraint in the verified DAE solver is defined by

$$\begin{aligned} u_{out}(t) &= u_C(t) \\ &= 1 + \frac{2}{3}\sqrt{3}\exp\left(-\frac{1}{2}t\right)\sin\left(\frac{1}{2}\sqrt{3}t\right). \end{aligned} \tag{62}$$

First, the case $\dot{u}_{in} = -50$ is evaluated. As shown in Fig. 6(a), the corresponding input voltage $u_{in}(t)$ in Fig. 6(b) is guaranteed to be inadmissible since the resulting and desired outputs of the dynamical system model do not overlap.

Second, enclosures for the set of admissible input signals are determined for $\dot{u}_{in} \in [-50\ ;\ 50]$, see Fig. 7. In this case, the time interval $t \in [0\ ;\ 0.5]$ is subdivided into time intervals with the fixed width 0.001. For each time interval, a piecewise linear control input is determined, where the following three cases are distinguished:

- The interval enclosure for $x_{out}(t)$ is completely included in the interval specifying the desired output. The corresponding input signal is guaranteed to be *admissible*.
- The interval enclosure for $x_{out}(t)$ is completely outside of the interval specifying the desired output. The corresponding input signal is guaranteed to be *inadmissible*.
- All remaining subintervals are *undecided*. They have to be investigated further by splitting of $[\dot{u}_{in}(t)]$.

The major advantages of this approach for the computation of input signals of a dynamical system are that

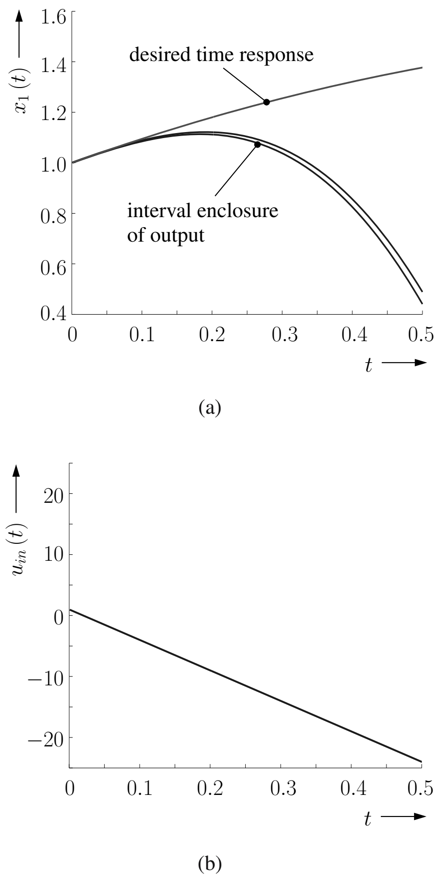


Fig. 6. Identification of inadmissible control sequences: (a) interval enclosure of $x_1(t)$, (b) interval enclosure of $u_{in}(t)$, $\dot{u}_{in} = -50$.

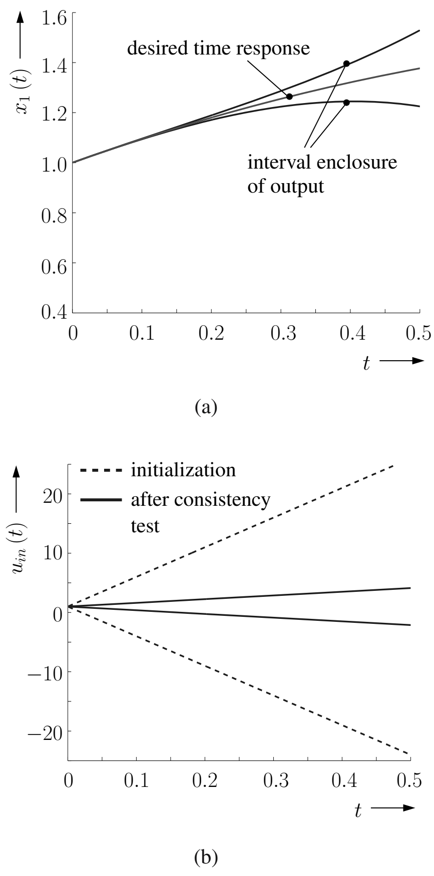


Fig. 7. Identification of admissible control sequences: (a) interval enclosure of $x_1(t)$, (b) interval enclosure of $u_{in}(t)$, $\dot{u}_{in} \in [-50\ ;\ 50]$.

the non-existence of appropriate solutions can be proven in a guaranteed way and that parameter uncertainties as well as tolerances for the admissible output signals can be handled without any modification of the computational procedure. Both types of uncertainties can be directly considered in the interval-based framework for the evaluation of sets of DAEs.

9. Conclusions and outlook on future research

In this paper, VALENCIA-IVP has been presented as an interval arithmetic solver for initial value problems for both ODEs and DAEs. In future work, extensions will be developed that help to deal directly with DAEs which are not given in a semi-explicit form. Furthermore, consistency tests as highlighted in Subsection 8.1, preconditioning strategies, and routines for automatic step size control will be investigated and implemented.

Regarding the computation of control sequences for dynamical systems with predefined outputs, two further extensions that will be considered in the future will broaden the applicability of VALENCIA-IVP. First, the step sizes used for switching the control inputs should be decoupled from the step size used for the integration of the set of DAEs. Second, coupling the routines described above for the computation of consistent control sequences with interval-based routines for the computation of optimal control strategies (Rauh and Hofer, 2009) will be investigated.

These two tasks will provide a possibility to further integrate verified techniques for the simulation of dynamical systems in frameworks for the design of robust and optimal controllers.

Acknowledgment

The authors would like to thank the reviewers for their detailed suggestions for improving of our paper. Furthermore, we would like to thank Ekaterina Auer, Faculty of Engineering, INKO, University of Duisburg-Essen, Germany, for her contributions to the development and implementation of VALENCIA-IVP.

Parts of this work were performed while the authors were with the Institute of Measurement, Control, and Microtechnology, University of Ulm, Germany.

References

Auer, E., Rauh, A., Hofer, E. P. and Luther, W. (2008). Validated modeling of mechanical systems with SMARTMOBILE: Improvement of Performance by VALENCIA-IVP, *Proceedings of the Dagstuhl Seminar 06021—Reliable Implementation of Real Number Algorithms: Theory and Practice, Dagstuhl, Germany,*, Lecture Notes in Computer Science, Vol. 5045, Springer-Verlag, Berlin/Heidelberg, pp. 1–27.

Bendsten, C. and Stauning, O. (2007). *FADBAD++*, *Version 2.1*, Available at: http://www.fadbad.com.

Berz, M. and Makino, K. (1998). Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models, *Reliable Computing* **4**(4): 361–369.

Cash, J. R. and Considine, S. (1992). An MEBDF code for stiff initial value problems, ACM Transactions on Mathematical Software (TOMS) **18**(2): 142–155.

Chua, L., Desoer, C. A. and Kuh, E. S. (1990). *Linear and Nonlinear Circuits*, McGraw-Hill, New York, NY.

Czechowski, P. P., Giovannini, L. and Ordys, A. W. (2006). Testing algorithms for inverse simulation, *Proceedings of the 2006 IEEE International Conference on Control Applications, Munich, Germany*, pp. 2607–2612.

de Swart, J. J. B., Lioen, W. M. and van der Veen, W. A. (1998). Specification of PSIDE, *Technical Report MAS-R9833*, CWI, Amsterdam, Available at: http://walter.lioen.com/papers/SLV98.pdf.

Deville, Y., Janssen, M. and van Hentenryck, P. (2002). Consistency techniques for ordinary differential equations, *Constraint* **7**(3–4): 289–315.

Eijgenraam, P. (1981). The solution of initial value problems using interval arithmetic, *Mathematical Centre Tracts No. 144*, Stichting Mathematisch Centrum, Amsterdam.

Galassi, M. (2006). GNU Scientific Library Reference Manual. Revised Second Edition (v1.8), Available at: http://www.gnu.org/software/gsl/.

Hairer, E., Lubich, C. and Roche, M. (1989). *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*, Lecture Notes in Mathematics, Vol. 1409, Springer-Verlag, Berlin.

Hairer, E. and Wanner, G. (1991). *Solving Ordinary Differential Equations II—Stiff and Differential-Algebraic Problems*, Springer-Verlag, Berlin/Heidelberg.

Hammersley, J. M. and Handscomb, D. C. (1964). *Monte-Carlo Methods*, John Wiley & Sons, New York, NY.

Hoefkens, J. (2001). *Rigorous Numerical Analysis with High-Order Taylor Models*, Ph.D. thesis, Michigan State University, East Lansing, MI, Available at: http://www.bt.pa.msu.edu/cgi-bin/display.pl?name=hoefkensphd.

Iavernaro, F. and Mazzia, F. (1998). Solving ordinary differential equations by generalized Adams methods: Properties and implementation techniques, *Applied Numerical Mathematics* **28**(2): 107–126.

Jaulin, L., Kieffer, M., Didrit, O. and Walter, É. (2001). *Applied Interval Analysis*, Springer-Verlag, London.

Keil, C. (2007). *Profil/BIAS*, *Version 2.0.4*, Available at: www.ti3.tu-harburg.de/keil/profil/.

Krawczyk, R. (1969). Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken, *Computing* **4**(3): 189–201, (in German).

Kunkel, P., Mehrmann, V., Rath, W. and Weickert, J. (1997). *GELDA: A Software Package for the Solution of General Linear Differential Algebraic Equations*, pp. 115–138, Available at: http://www.math.tu-berlin.de/numerik/mt/NumMat/Software/GELDA/.

Lin, Y. and Stadtherr, M. A. (2007). Deterministic global optimization for dynamic systems using interval analysis, *CD-Proceedings of the 12th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN 2006, Duisburg, Germany*, IEEE Computer Society, Los Alamitos, CA.

Moore, R. E. (1966). *Interval Arithmetic*, Prentice-Hall, Englewood Cliffs, New Jersey, NY.

Nedialkov, N. S. (2007). Interval tools for ODEs and DAEs, *CD-Proceedings of the 12th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN 2006, Duisburg, Germany*, IEEE Computer Society, Los Alamitos, CA.

Nedialkov, N. S. and Pryce, J. D. (2008). *DAETS—Differential-Algebraic Equations by Taylor Series*, Available at: http://www.cas.mcmaster.ca/~nedialk/daets/.

Petzold, L. (1982). A description of DASSL: A differential/algebraic systems solver, *IMACS Transactions on Scientific Computation* **1**: 65–68.

Rauh, A. (2008). *Theorie und Anwendung von Intervallmethoden für Analyse und Entwurf robuster und optimaler Regelungen dynamischer Systeme*, Fortschritt-Berichte VDI, Reihe 8, Nr. 1148, Ph.D. thesis, University of Ulm, Ulm, (in German).

Rauh, A. and Auer, E. (2008). Verified simulation of ODEs and DAEs in VALENCIA-IVP, *Proceedings of the 13th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN 2008, El Paso, TX, USA*, (under review).

Rauh, A., Auer, E., Freihold, M., Hofer, E. P. and Aschemann, H. (2008). Detection and reduction of overestimation in guaranteed simulations of hamiltonian systems, *Proceedings of the 13th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN 2008, El Paso, TX, USA*, (under review).

Rauh, A., Auer, E. and Hofer, E. P. (2007a). VALENCIA-IVP: A comparison with other initial value problem solvers, *CD-Proceedings of the 12th GAMM-IMACS International*

Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN 2006, Duisburg, Germany, IEEE Computer Society, Los Alamitos, CA.

Rauh, A., Auer, E., Minisini, J. and Hofer, E. P. (2007b). Extensions of VALENCIA-IVP for reduction of overestimation, for simulation of differential algebraic systems, and for dynamical optimization, *Proceedings of the 6th International Congress on Industrial and Applied Mathematics, Minisymposium on Taylor Model Methods and Interval Methods—Applications, PAMM, Zurich, Switzerland*, Vol. 7(1), pp. 1023001–1023002.

Rauh, A. and Hofer, E. P. (2009). Interval methods for optimal control, *in* A. Frediani and G. Buttazzo (Eds.), *Proceedings of the 47th Workshop on Variational Analysis and Aerospace Engineering 2007, Erice, Italy*, Springer-Verlag, New York, NY, pp. 397–418.

Rauh, A., Minisini, J. and Hofer, E. P. (2009). Towards the development of an interval arithmetic environment for validated computer-aided design and verification of systems in control engineering, *Proceedings of the Dagstuhl Seminar 08021: Numerical Validation in Current Hardware Architectures, Dagstuhl, Germany,* Lecture Notes in Computer Science, Vol. 5492, Springer-Verlag, Berlin/Heidelberg, pp. 175–188.

Röbenack, K. (2002). On the efficient computation of higher order maps $ad_f^k g(x)$ using Taylor arithmetic and the Campbell-Baker-Hausdorff formula, *in* A. Zinober and D. Owens (Eds.), *Nonlinear and Adaptive Control*, Lecture Notes in Control and Information Science, Vol. 281, Springer, London, pp. 327–336.

Appendix

Krawczyk method. Like ordinary Newton techniques, the Krawczyk iteration is used to solve algebraic equations $h(x) = 0$. The major difference is that both the initial guess and the computed solution are provided as interval boxes. In the case of convergence, the Krawczyk iteration guarantees the existence of the solution of $h(x) = 0$ in the interval enclosure $[x_{zero}]$.

After initialization with the initial guess $[x_{zero}^{(0)}]$, the Krawczyk iteration

$$\begin{aligned}&\left[k\left(\left[x_{zero}^{(\kappa)}\right]\right)\right]\\&:= x_m^{(\kappa)} - Y \cdot h\left(x_m^{(\kappa)}\right)\\&\quad + \left(I - Y \cdot \left.\frac{\partial h}{\partial x}\right|_{x=\left[x_{zero}^{(\kappa)}\right]}\right) \cdot \left(\left[x_{zero}^{(\kappa)}\right] - x_m^{(\kappa)}\right)\end{aligned} \tag{63}$$

with

$$Y^{-1} \in \left.\frac{\partial h}{\partial x}\right|_{x=\left[x_{zero}^{(\kappa)}\right]} =: [h'] \tag{64}$$

is computed. Here, $x_m^{(\kappa)}$ is the midpoint of the interval-vector $[x_{zero}^{(\kappa)}]$ and I stands for an identity matrix of appropriate dimensions. The matrix Y^{-1} is usually defined by $\mathrm{mid}([h']) = \frac{1}{2}\left(\underline{h'} + \overline{h'}\right)$ with

$$[h'] := \left.\frac{\partial h}{\partial x}\right|_{x\in[x_{zero}^{(\kappa)}]} . \tag{65}$$

The required point matrix Y is computed by inverting $\mathrm{mid}([h'])$. If this matrix is singular, Y^{-1} can be redefined as an arbitrary point matrix contained in $[h']$.

After each iteration step κ, the result $[k([x_{zero}^{(\kappa)}])]$ is compared with $[x_{zero}^{(\kappa)}]$. If the relation

$$\left[k\left(\left[x_{zero}^{(\kappa)}\right]\right)\right] \subset \left[x_{zero}^{(\kappa)}\right] \tag{66}$$

holds, the desired solution $[x]$ is guaranteed to be included in the interval $[x_{zero}^{(\kappa)}]$. Hence, for the next iteration step $\kappa + 1$,

$$\left[x_{zero}^{(\kappa+1)}\right] := \left[k\left(\left[x_{zero}^{(\kappa)}\right]\right)\right] \tag{67}$$

can be used. If (66) is not true, there are two cases to be distinguished. If a non-empty intersection

$$\left[k\left(\left[x_{zero}^{(\kappa)}\right]\right)\right] \cap \left[x_{zero}^{(\kappa)}\right] \neq \emptyset \tag{68}$$

with

$$\left[k\left(\left[x_{zero}^{(\kappa)}\right]\right)\right] \not\subset \left[x_{zero}^{(\kappa)}\right] \tag{69}$$

exists, the initial guess for the interval-vector $[x_{zero}^{(\kappa)}]$ has to be inflated. If

$$\left[k\left(\left[x_{zero}^{(\kappa)}\right]\right)\right] \cap \left[x_{zero}^{(\kappa)}\right] = \emptyset \tag{70}$$

holds for the intersection, the solution $[x]$ is certainly not contained in $[x_{zero}^{(\kappa)}]$. Then, a new initialization has to be chosen.

Andreas Rauh received his diploma degree in electrical engineering and information technology from Technische Universität München, Munich, Germany, in 2001 and his Ph.D. degree (Dr.-Ing.) from the University of Ulm, Germany, in 2008. His research interests include state and parameter estimation for stochastic and set-valued uncertainties, verified simulation of nonlinear uncertain systems, nonlinear, robust and optimal control, interval methods for ordinary differential equations as well as differential-algebraic systems. Currently, he is with the Chair of Mechatronics, University of Rostock, Germany, as a post-doctoral researcher.

Michael Brill received his diploma degree in electrical engineering with special interest in automation and energy technology from the University of Ulm, Germany, in 2007. His diploma thesis was performed at the Institute of Measurement, Control, and Microtechnology. It describes an extension to VALENCIA-IVP which allows verified solving of sets of differential-algebraic equations directly without rearranging to sets of ordinary differential equations. Currently, he works towards his Ph.D. for an onboard algorithm predicting the ageing of automotive Li-Ion battery packs.

Clemens Günther received his diploma in electrical engineering from the University of Ulm, Germany, in 2008. He wrote his thesis at the Institute of Measurement, Control, and Microtechnology. In his work, he concentrated on interval methods for control unit design for technical systems. He developed an algorithm for verified computation of control strategies for uncertain systems based on VALENCIA-IVP. Clemens Günther currently works at Zentrum für Sonnenenergie- und Wasserstoff-Forschung, Ulm. There, he conducts research into modeling and simulating electrochemical accumulators, like lithium ion accumulators, to receive a Ph.D. degree.

Received: 22 September 2008
Revised: 15 December 2008