Uses of Verified Methods for Solving Non-Smooth Initial Value Problems[*]

Ekaterina Auer, Stefan Kiel

Computer and Cognitive Sciences (INKO),
University of Duisburg-Essen,
Duisburg, Germany

auer@inf.uni-due.de
kiel@inf.uni-due.de

Abstract

Many system types in engineering require mathematical models involving non-differentiable or discontinuous functions. Such system models are often sensitive to round-off errors. Their parameters might be uncertain due to impreciseness in measurements or lack of knowledge. Therefore, interval methods represent a straightforward choice for verified analysis of such systems. However, application of existing interval methods to real-life scenarios is challenging, since they might provide overly conservative enclosures of exact solutions. In this paper, we analyze a simple method to obtain meaningful solution enclosures. First, we identify important types of non-smooth applications along with their corresponding solution definitions. After that, we provide an overview of existing methods for verified enclosure of exact solutions to non-smooth IVPs. Next, we introduce a simple enclosure strategy which relies on basically the same techniques as in the smooth case. Finally, we demonstrate the applicability of the simple method using several examples, and compare the results to those produced by the existing techniques.

Keywords: interval methods, non-smooth systems, initial value problems
AMS subject classifications: 65G40

1 Introduction

A large number of applications from the theory of automatic control, mechanics, or electrical engineering are represented by mathematical models that depend on discontinuous or non-differentiable functions. Such situations occur, for example, when engineers describe systems with friction, take into account saturation effects in quantities of interest, or simply express naturally arising conditions such as non-positivity of

[*]Submitted: February 10, 2013; Revised: December 10, 2013; Accepted: December 23, 2013.

variables. The task becomes especially complicated if non-smooth initial value problems (IVPs) for e.g. ordinary differential equations (ODEs) are considered. Here, even the definition of the solution might depend on the application at hand. Solving such problems is often additionally impeded by uncertainty in parameters. Besides, the solution might be sensitive to numerical errors. One possibility to deal with these difficulties is the use of verified methods both during the modeling and the simulation stages.

Verified methods [24] provide a guarantee that results obtained on a computer are consistent with the formal model developed for the considered real-life system. The application of existing interval methods to real-life scenarios is challenging since they might provide overly conservative enclosures of exact solutions (*exact* in a predefined sense). Even in the case of a few jump discontinuities in the right side of an IVP, where the solution is not differentiable in just several switching points, the accuracy after encountering such a point might be poor and, consequently, the resulting enclosures might be too wide [34]. This is probably the reason for the relatively little attention non-smooth problems have received in recent decades, whereas the verified solution of smooth IVPs has been extensively explored [4, 6, 17, 21, 22, 28, 29, 32].

There are two general directions in describing a discontinuous problem. The first possibility, known as *noncausal modeling*, is to formulate it as a single system of ordinary or implicit differential (algebraic) equations/inequalities (ODEs or DAEs, respectively) that changes, for example, in dependence on zeros of a certain continuous function [1]. The second possibility is to represent the problem as a kind of a graph or an automaton with different ODEs as vertices and logical conditions for jumps as edges [18] (*causal* or *graphical modeling*). The main simulation methods from the area of usual numerics are based on event-driven [39] or time-stepping [3] schemes as well as on constrained nonlinear programming [1], formal model checking, and formal verification [18]. The noncausal modeling is sometimes viewed as a special case of the causal. However, the first alternative is used more widely if a physical system is modeled whereas the second one is more common in the area of control.

In traditional non-smooth theory, awareness of methods with result verification, for example, interval analysis, is relatively low. However, they can be of great help for solving differential equations with convex and closed set-valued right sides. Therefore, it is interesting to study how traditional solution methods can be aided or covered completely by those from the interval analysis.

We summarize the most important concepts from the area of non-smooth modeling and simulation in Section 2. In particular, we define the type of solution definition we are interested in and the class of problems it can cover. After that, we summarize existing techniques for the calculation of verified enclosures of solutions to non-smooth IVPs in Section 3. Next, we focus our considerations on a special case in which the switching points are known a priori in a certain sense. For this situation, we describe, in Section 4, a simple method to solve non-smooth IVPs using basically the same techniques as in the smooth case we introduced in [2]. We touch upon the proposed approach and its implementation for the solver ValEncIA-IVP [32]. Finally, we explore applicability of the method from the numerical point of view using a number of examples from physics and control theory. The results are compared with those from the existing verified methods where possible. A discussion of the results and an outlook on our future work are in Section 5.

2 Main Concepts of Non-smooth Theory

In this Section, we summarize typical possibilities for formalization of non-smooth problems, along with the appropriate solution definitions from the point of view of traditional theory. Where possible, the corresponding interval-based reformulations are provided. In this overview, we concentrate mostly on initial value problems for ODEs.

Non-smooth or discontinuous IVPs arise naturally in many practical applications [1, 15]. In mechanics, they describe, for example, systems with friction, with impacts, with piecewise contact laws or with hysteresis; in electrical engineering, electrical circuits with (ideal) diodes; in control engineering, sliding or switching control systems as well as a number of optimal control laws; in biology, systems with instantaneous switches or with hysteresis. In addition, non-smooth representations are useful in economics, hydraulic circuits, material science, and many other areas. The mathematical formalisms and solution methods for problems in different fields might be very similar. For example, the complementarity condition between the current across an ideal diode and its voltage (electrical circuits) is similar to the relation between the contact force and the distance between the system and an obstacle in unilateral mechanics [1]. The two characteristics in question should be orthogonal to each other and non-negative, a condition which is non-smooth and multivalued.

What is often overlooked or neglected is the fact that non-smoothness might arise in every research area relying on numerical computations when scientists try to ensure numerical stability in their programs. From the point of view of implementation, non-smoothness can be caused simply by the presence of `IF-THEN-ELSE` or `SWITCH` statements on variables in a program code. For example, the exponential camelback function

$$f_{ecb}(x) = a_1 e^{-w_1(x-b_1)^2} + a_2 e^{-w_2(x-b_2)^2}$$

is often used for modeling purposes. If it is employed for describing muscle activation as in [40], the parameters $a_1, a_2, w_1, w_2, b_1, b_2$ are selected so the function values lie between zero and one. To avoid errors caused by floating-point overflow or underflow, we might want to produce the following computer code:

```
v=f_ecb(5);
if (v>1)   v=1;
if (v<=10^-6) v=0;
```

Obviously, the function in the code is discontinuous at least at those x^* for which $f_{ebc}(x^*) = 10^{-6}$. If it is used as a part of the equations of motion for a flexion/extension two-joint subsystem model of a hip or a knee, a discontinuous IVP results. This kind of discontinuity is not as obvious as that arising in, for example, systems with impacts, since it is often hidden in code. Although floating-point based IVP solvers might be rather accurate in treating such problems as smooth ones, there is no guarantee of correctness for the obtained results. Since verified IVP solvers rely on derivatives in their algorithms, they would immediately encounter a conceptual problem in this situation.

As mentioned in the introduction, there are two conceptual modeling possibilities for non-smooth problems. For more information on the causal case, see [18]. For the noncausal representation, researchers assume that the system is given in terms of analytical expressions, for example,

$$\dot{x}(t) = f(t, x(t)),\ x(t_0) = x_0\ , \tag{1}$$

with a possibly discontinuous function f. Discontinuities might be expressed in different ways, for instance, as jumps depending on zeros of a certain switching function. A common formalism to describe and understand such systems in the traditional theory is that of *differential inclusions* (DI) [8]

$$\dot{x}(t) \in F(t, x(t)), \quad t \in [t_0, t_{\text{end}}], \quad x(t_0) = x_0 \ , \tag{2}$$

where F is a set-valued map $\mathbb{R} \times \mathbb{R}^n \mapsto S(\mathbb{R}^n)$, with $S(\mathbb{R}^n)$ a set of subsets of $\mathbb{R}^n$. The map F is usually chosen in such a way to coincide with f in its areas of smoothness. Geometrically, F might represent convex (or non-convex) bounded (or unbounded) sets depending on the application at hand. In each case, a different kind of analysis is required. A solution $x(t)$ to this problem should satisfy the inclusion in (2) almost everywhere on $[t_0, t_{\text{end}}]$. Different applications dictate different further requirements for $x(t)$. In this paper, we are interested in an absolutely continuous solution $x(t)$, in particular, we rule out discrete jumps in states.

According to the type of F, two important classes of DIs can be discerned [1]: Lipschitzian and upper semi-continuous. An autonomous DI of the form (2), where F depends only on states x, is called *Lipschitzian*, if the sets $F(x)$ are closed and convex for each x and the map $F(x)$ is Lipschitz, that is,

$$F(x_1) \subset F(x_2) + l||x_1 - x_2|| \cdot \{y \mid ||y|| \leq 1\} \tag{3}$$

for all $x_1, x_2 \in \mathbb{R}^n$ and l a positive constant. Each function $x(t)$ satisfying (2) almost everywhere with the mapping F defined as above is a solution of (2). Each absolutely continuous function $x(t)$ for which (1) holds is a solution of the Lipschitzian DI. Conversely, there always exists a solution of the DI which is also the solution of (1).

A set-valued map $F(x)$ is called *upper hemicontinuous* (in some works, upper/outer semi-continuous) in x, if for any open neighborhood $N \supset F(x)$ there exists a neighborhood $M \ni x$ such that $F(M) \subset N$. A DI of the form (2) is called *upper semi-continuous* if F is upper hemicontinuous and the sets $F(x)$ are closed and convex for all x. If the condition $||F(x)|| \leq c(1 + ||x||)$ for some $c > 0$ is fulfilled additionally, then there exists an absolutely continuous solution to the DI on the positive real time axis [1].

A special case of upper semi-continuous DIs are *Filippov's systems.* The intention behind them is to formulate the problem in such a way to ensure the existence of solutions and their compliance with solutions to IVPs with continuous right sides in the areas of smoothness. In the "simplest convex case", the definition is that for each (t, x^*), x^* a point of discontinuity, $F(t, x^*)$ is the smallest convex closed set containing all the limit values of the function $f(t, x)$ where t is fixed, (t, x) is not a point of discontinuity, and $x \to x^*$ [8]. There always exists an absolutely continuous solution to a Filippov's DI. The solution to (1) is then the solution to (2) with F defined according to Filippov.

Lipschitzian DIs cover a smaller class of problems than Filippov's DIs. It can be shown [1] that if F satisfies a one-sided Lipschitz condition, the solution to (2) is unique.

If F maps to the set of intervals $\mathbb{IR}^n$ and can be expressed as a continuous function with interval parameters, the task of finding solutions to a DI is related to a continuous initial value problem in its interval formulation. Here, we replace the inclusion in (2) by the equality sign and search for a functional tube

$$\boldsymbol{x}(t) = \left\{ x(t) = x_0 + \int\limits_0^t \tilde{f}(x(s))ds \;\Big|\; \text{for all } x_0 \in \boldsymbol{x}_0 \right\} \tag{4}$$

as the solution. (Note that problems with interval parameters can be reformulated into problems where only the initial values $\boldsymbol{x}_0$ are intervals, so $\tilde{f}$ represents such a reformulated function for F.) The sets $F(x)$ are then closed and convex and the condition (3), for example, can be replaced by the interval Lipschitz condition

$$F(x_1) - F(x_2) \subset \boldsymbol{L}(x_1 - x_2),\ \boldsymbol{L} \in \mathbb{IR}^n$$

to obtain an interval-based reformulation. If $\tilde{f}$ is (at least) differentiable[1], such an enclosure of the solution set can be computed using smooth methods of interval analysis [6, 17, 27]. The numerical algorithms in these works provide an enclosure guaranteed to contain the exact solution. If $\tilde{f}$ is only Lipschitzian, the method proposed in Section 4 can be used.

A further important class of non-smooth problems are *ODEs with discontinuities*:

$$\dot{x}(t) = f(t, x(t)) = \begin{cases} f_1(t, x(t)),\ g(t, x(t)) < 0 \\ f_2(t, x(t)),\ g(t, x(t)) > 0 \end{cases} \tag{5}$$

with $x(t_0) = x_0$, where $f_1(t, x)$, $f_2(t, x)$, and $g(t, x)$ are smooth. Points x^* such that $g(t, x^*) = 0$ for some t are called switching points. Both f_1 and f_2 are considered to be bounded in x^*. Such systems are a special case of Filippov's systems. This situation is well-explored theoretically [23, 39], in particular, under presence of bounded additive uncertainty [30] and in the interval case [35]. The main appraoch is to compute

$$\dot{g}_1(t, x) := \frac{\partial g}{\partial t} + \frac{\partial g}{\partial x} f_1(t, x), \tag{6}$$

$$\dot{g}_2(t, x) := \frac{\partial g}{\partial t} + \frac{\partial g}{\partial x} f_2(t, x)\ . \tag{7}$$

If the *transversality condition* holds, that is, if $\dot{g}_1(t, x) > 0$ and $\dot{g}_2(t, x) > 0$, then the solution crosses over the switching surface $\{(t, x) : g(t, x) = 0\}$ from the area $G_1 = \{g(t, x) < 0\}$ to the area $G_2 = \{g(t, x) > 0\}$. A practically interesting case occurs if $\dot{g}_1(t, x) < 0$ and $\dot{g}_2(t, x) > 0$. The solutions in G_1 and G_2 both run into the switching surface and the solution to (5) has to stay there[10]. The simple convex definition mentioned earlier allows for sliding along such switching surfaces. For the problem (5), this definition can be reduced to

$$F(t, x) = \{\alpha f_1(t, x) + (1 - \alpha) f_2 \mid \alpha \in [0, 1]\} \tag{8}$$

for (t, x) on the switching plane.

In addition to bounded/unbounded differential inclusions, such formalisms as linear complementarity systems, Moreau's sweeping process, unilateral DIs, evolution variational inequalities, differential variational inequalities, projected dynamical systems and others are to be found in literature. As stated in e.g. [1, 18], introducing large general classes of descriptions is useful only in a limited way, and narrow classes have to be defined to obtain accurate results. In this paper, we will focus on convex closed DIs (including ODEs with discontinuities) and point out where an interval-based reformulation of the problem description is possible. Some of the other mentioned concepts can also be covered, since they are related. Example 2.43 for evolution variational inequalities from [1] illustrates such a correlation. There, the dynamics of a system with Coloumb and viscous friction are modeled initially by the inclusion

$$m\ddot{q}(t) + c\dot{q}(t) + kq(t) \in -\partial\varphi(\dot{q}(t)) = -\mu \cdot \operatorname{sign}(\dot{q}) \tag{9}$$

[1] In general, better enclosures can be obtained if the right side is differentiable up to higher orders.

with $\varphi(\dot{q}) = \mu|\dot{q}|$, m, c, k, μ positive coefficients, and the set-valued sign function

$$\operatorname{sign}(x) = \begin{cases} -1, & x < 0, \\ [-1, 1]\,, & x = 0, \\ 1, & x > 0. \end{cases} \tag{10}$$

The DI above is reformulated in [1] as a variational inequality and as an evolution variational inequality. In Section 4 of this paper, we will suggest a method to solve the following interval formulation of this problem:

$$\begin{cases} \dot{\boldsymbol{x}}_1 &= \boldsymbol{x}_2, \\ \dot{\boldsymbol{x}}_2 &= -\frac{\mu}{m} \cdot \operatorname{sign}(\boldsymbol{x}_2) - \frac{c}{m}\boldsymbol{x}_2 - \frac{k}{m}\boldsymbol{x}_1, \end{cases} \tag{11}$$

with the sign function defined in the same way as above and $x_1 = q$, $x_2 = \dot{q}$. Here, the solution is understood as given in Eq. (4). It contains a/the exact solution to the problem in the sense of definitions mentioned earlier. We restrict our considerations to interval enclosures at discrete points of time $\{t_k = t_{k-1} + h_k \mid k = 1, \ldots, n\}$, where $t_n = t_{\text{end}}$, h_k the stepsize. The corresponding (continuous) tube is usually obtained using a Taylor expansion with the suitable interval Taylor coefficients and an enclosure of the error term over each time interval $[t_j, t_{j+1}]$ [16], page 45. In our case, this is not possible for intervals containing switching points. Here, a coarse zero-order enclosure must be used. An alternative is an enclosure relying on the mean value theorem with the generalized derivative definition as found in [13, 25] or as given in Section 4.

In the traditional theory, there are more theoretical and more practical approaches to describing non-smooth systems. Some researchers choose to focus on qualitative analytical characteristics such as (non-smooth) Lyapunov exponents, Conley's index, the Kolmogorov-Arnold-Moser theory, or Melnikov's theory [3, 15]. Another group of researchers is explicitly interested in devising accurate numerical algorithms for characterizing the solutions [1, 23, 39]. Direct numerical simulation methods (time-stepping or event-driven) can (or should) be supplemented by the so-called path-following methods to accurately compute bifurcation points and unstable invariant sets [3].

3 Overview of the Existing Verified Methods for Non-smooth IVPs

As pointed out before, interval methods [24] offer a natural way of taking into account bounded, purely epistemic uncertainty. Additionally, in the case of smooth dynamics, they provide a guarantee that the resulting numerical enclosure contains the exact solution to the system model being considered. Their main drawback is possible overestimation, that is, conservative enclosures which are too wide to give any information about the system's behavior. In this section, we overview briefly the main existing works on result verification for non-smooth problems and summarize their characteristic features in Table 1. We consider both causal and noncausal approaches.

A simple verified method, which we described in detail in [2], and which is recapitulated in the next section, is shown in the last column. From the table, it can be observed that this approach needs less strict requirements than most of the other methods to produce a verified enclosure, at the cost of representing the right hand side in the form (15) and of wider enclosures.

Table 1: Summary of different verified methods for non-smooth IVPs. Our method from the next Section is denoted by **A**. The letters c and n signify causal (automaton) and noncausal (mathematical) representation, rsp.; + and - indicate the availability or absence of a feature, rsp., o shows that a property does not apply; p stands for piecewise continuous; L means Lipschitz, h, m and l stand for *high*, *medium* and *low*, rsp.

	Method								
Property	[7]	[11]	[12]	[20]	[26]	[31]	[33]	[35]	A
representation	c	c	c	n	c	c	c	n	n
conditions for f	p	p	p	L	p	p	p	p	p^{**}
conditions for f_i	$\mathcal{C}^q$	$\mathcal{C}^{1^*}$	L	o	$L^*/\mathcal{C}^q$	$\mathcal{C}$	$\mathcal{C}^q$	$\mathcal{C}^q$	$\mathcal{C}^1$
use of transversality	-	-	-	-	+	-	-	+	-
analytical solution	-	+/-	-	-	+/-	-	-	-	-
continuousness of solution	o	p	p	+	p	o	p	+	+
sliding solutions	-	-	-	-	-	-	-	+	-
use of verified IVPS	+	+	+	o	+	+	+	+	+
constraint prop.	+	+	+	-	-	+	-	-	-
verified equation solvers	+	-	+	+	+	-	-	+	-
formal model checkers	+	-	o	o	o	+	-	o	o
verified solution enclosure	-	+	+	-	+	-	+	+	+
user involvement	l	l	l	l	l	l	h	l	l
tightness	o	h	o	o	h	o	m	h	m

[*] f_i should be chosen so the IVP will have an analytical solution if no verified integration is used (e.g., linear).
[**] f_i should be representable in the form (15).

In the works [34, 35, 36], the goal is to find an enclosure of the exact solution of (5). The exact solution is defined as a continuous function $x(t) : I \subset [t_0, t_{\text{end}}] \mapsto \mathbb{R}^n$ satisfying (5), if the zeros of the switching function $g(t, x(t))$ are isolated, $\dot{x}(t) = f(t, x(t))$ everywhere except possibly on the set of all zeros of the switching function, and the initial condition $x(t_0) = x_0$ holds. The solution runs through the switching point at t^* (the point where the solution $x(t)$ intersects the switching surface $\{g(t, x(t)) = 0\}$). If the transversality conditions are satisfied, the solution is unique and can be continued from G_1 to G_2 according to the formula

$$x(t^+) \in \boldsymbol{x}^- + s\boldsymbol{f}^- + [0, s](\boldsymbol{f}^+ - \boldsymbol{f}^-) + \boldsymbol{z}^- + \boldsymbol{z}^+ =: \boldsymbol{x}^+,$$

where $t^* \in [t^-, t^+]$ is an enclosure of the current switching point (obtained with e.g. Newton's method), $\boldsymbol{x}^-$ denotes the enclosure of the exact solution $x(t^-)$ in the area of smoothness G_1 (obtained with a smooth IVP method), t^+ is situated to the right of the switching point t^*, $\boldsymbol{x}^+$ is the enclosure of the solution there, $s = t^+ - t^-$, $\boldsymbol{z}^-$ and $\boldsymbol{z}^+$ are the enclosures of the exact local discretization errors, $\boldsymbol{x}^* := \boldsymbol{x}^- + [0, s]\boldsymbol{f}^- + \boldsymbol{z}^-$, and $\boldsymbol{f}^- := f_1(t^-, \boldsymbol{x}^-)$, $\boldsymbol{f}^+ := f_2([t^-, t^+], \boldsymbol{x}^*)$ are interval evaluations of the smooth functions f_1 and f_2.

This does not yet cover sliding solutions where $\dot{g}_1(t, x) < 0$, $\dot{g}_2(t, x) > 0$ and the zeros of the switching function are not isolated. To take into account such situations, the author resorts to the concept of Filippov's systems, and defines a solution for which the DI (2) holds except possibly on a set of isolated points. Here, the set-valued F is supposed to coincide with f_1 in the area G_1, f_2 in the area G_2, and is allowed to contain sets of higher cardinality at the switching surfaces where $g(t, x(t)) = 0$. To solve the problem, the author considers DIs (2) with the right sides of the form (8), where

$$\alpha(t, x) := \frac{\dot{g}_2(t, x)}{\dot{g}_2(t, x) - \dot{g}_1(t, x)} \in [0, 1] \tag{12}$$

is continuous. The function

$$f_0 := \alpha(t, x) f_1(t, x) + (1 - \alpha(t, x)) f_2(t, x) \tag{13}$$

is also continuous, that is, there exists at least one classical solution to $\dot{x} = f_0(t, x)$, $x(t^*) = x^*$ which is also a solution to (2). Under the assumption that f_0 is continuously differentiable, the same approach as for the valid transversality conditions can be applied to the modified IVP.

In [26], the method described above is applied to non-smooth hybrid systems in combination with the smooth IVP solver VNODE [27]. Further verified approaches for the causal representation are described in [11, 31, 33]. In [33], the authors study dynamical systems consisting of l different smooth models, which are interpreted as discrete states of the overall non-smooth model for the real world system. A transition from the currently active state S_i to the state S_j, $i, j = 1, \ldots, l$, takes place if the condition $T_i^j(x, u)$, which depends on the solution and the control input, becomes active. The goal is to compute guaranteed enclosures of solutions to the non-smooth model at discrete points of time. For this purpose, the authors suggest extending smooth IVP solving routines from interval analysis with a technique to detect all possible points of time at which transition conditions T_i^j are activated. To provide tight enclosures of the state variables, it is necessary to detect as soon as possible that one of the states from $\mathcal{S}$ is not active at a given point of time. In essence, the authors first detect all possible points of time at which transition conditions $T_i^j(x, u)$ are activated by enclosing the sets of reachable state variables using rough bounds provided by Picard iteration. After that, they determine tighter enclosures of state variables by a Taylor series method (of zero order, if a switching point is inside the current time interval) and so eliminate states S_i that can be deactivated. In the current implementation from [33], the transition/deactivation conditions T_i^j, D_i^j as well as the appropriate sets for the right sides of IVPs have to be specified manually, which, in general, need not be so, cf. [7, 31]. The approach is implemented in MATLAB.

In [7, 31], it is formally verified that a hybrid system given as an automaton does not reach a set of states marked as unsafe. The authors rely on formal verification algorithms in combination with verified smooth IVP solvers such as VNODE-LP [27].

In [12], the problem is formulated as a hybrid constraint system, which consists of instantaneous constraints, continuous constraints on trajectories, and guard constraints on states causing discrete changes. To find an enclosure of the solution to such systems, the authors combine an interval IVP solver and a constraint programming technique which helps to reduce the width of the obtained enclosures.

In [11], a hybrid system model checker is developed which can handle dynamics expressed as a combination of polynomials, exponentials, and trigonometric functions. It conservatively overapproximates the set of reachable states by using interval methods as a tool for computing the flow successors of a given region.

A different approach that combines verified and traditional algorithms for IVPs (1)[2] with Lipschitz continuous right sides is that from [20]. The authors adapt an implicit Runge-Kutta (IRK) method of order s to non-smooth problems. First, they solve

[2] with $\frac{d(M \cdot x(t))}{dt}$ instead of just $\frac{dx(t)}{dt}$

the nonlinear system of equations associated with IRK using a generalized Krawczyk algorithm to obtain an enclosure $\boldsymbol{x}_k$ of its exact solution. After that, they prove that the error of the approximation to the IVP solution obtained in the second step of the IRK is smaller than a certain weighted sum of the maximal widths of $\boldsymbol{x}_k$. Here, the midpoints of $\boldsymbol{x}_k$ are used to compute the approximation.

There is a strong connection between non-smooth IVPs and non-smooth optimization. The works [13, 25] study from the verified point of view the relationships between different generalizations of the derivative for non-smooth functions and their application to optimization problems and in particular to solving systems of equations. In [37], the concept of slopes is implemented and extended to second order.

4 A Simple Verified Approach

As pointed out in [13] with respect to non-smooth optimization, "simplicity is a major advantage of treating [...] non-smooth problems with the same techniques as smooth problems". This is also true in the area of solving IVPs and provides a contrast between our approach and those outlined in the previous Section. The generalization of the derivative described in detail in [2] is not the same as in the formulas from [13], since it combines symbolic and automatic differentiation techniques while factoring in the development point of the mean value theorem similarly to slopes to improve the resulting enclosures. Moreover, our technique is different from the implementation proposed in [37], since we do not overload all variables with the data type for slopes (or derivatives, in our case) to obtain an enclosure of a function in a bottom-up approach, but compute it instead in a top-down way.

In this Section, we briefly summarize our method from [2], give details on its implementation and usage, and test it with many practically relevant examples with and without known exact solutions. Where possible, we compare the results with existing ones.

4.1 Main Definitions and the Method

Let the right side $f : D \subset \mathbb{R}^n \to \mathbb{R}^n$, where D is open, of the autonomous IVP with uncertain initial values from the interval $\boldsymbol{x}_0 \in \mathbb{IR}^n$

$$\dot{x} \quad = \quad f(x), \quad x(0) \in \boldsymbol{x}_0 \ , \tag{14}$$

be available in its algorithmic representation [38], that is, as a sequence of elementary operations and their compositions. In the usual understanding, the set S_{EO} of elementary operations consists of binary operations $+, -, *, /$ and unary *elementary functions* such as trigonometric ones. In our setting, we allow a further type of elementary function, namely, unary non-smooth ones [2]:

$$\varphi(y) = \begin{cases} \varphi_0(y) & \text{for } c_{-1} = -\infty < y < c_0, \\ \varphi_1(y) & \text{for } c_0 < y < c_1, \\ \cdots & \cdots \\ \varphi_{L-1}(y) & \text{for } c_{L-2} < y < c_{L-1}, \\ \varphi_L(y) & \text{for } c_{L-1} < y < c_L = +\infty \ . \end{cases} \tag{15}$$

Here, $c = (c_0, \ldots, c_{L-1})$ are switching points in terms of the scalar input variable y. The subfunctions $\varphi_0, \ldots, \varphi_L$ should be continuous, differentiable, and bounded in the whole definition domain. Non-smooth operations depending on more than one variable cannot be represented in this setting. However, if a non-smooth function depending on more than one variable can be constructed of unary non-smooth ones, then the definition in Eq. (15) covers it (e.g. $f(x_1, x_2) = |x_1| + x_1 \cdot \mathrm{sign}(x_2)$ and similar situations). The formulation (5) can cover the situation where the elementary non-smooth function depends on two or more variables. Yet, it is difficult to represent, for example, the dead time

$$f(x) = \begin{cases} -h, & x < -x_+ \\ 0, & -x_+ < x < x_+ \\ h, & x_+ < x \ . \end{cases} \tag{16}$$

(with h, x_+ constants) in the form (5) whereas (15) covers it naturally.

If a function φ_i is smooth, we understand its interval extension $\varphi_i(\boldsymbol{x})$ over $\boldsymbol{x} \in \mathbb{IR}$ as the natural interval extension. An interval extension of a function φ from (15) over $\boldsymbol{x}$ is given in [2] along with a possible interval extension for its generalized derivative in both the continuous and the discontinuous case. From these definitions, we obtain set-valued functions with closed and convex sets. The mean value theorem holds for the so defined derivative extension.

We use these notions in combination with the software ValEncIA-IVP to solve non-smooth problems of the form (14). We chose this particular software because it needed only the Jacobian of the right side of the ODE to provide guaranteed solution enclosures. We understand the term *solution* in the sense defined in [35] (cf. Section 3), if the right hand side of the IVP is non-smooth in x. Such solutions are known to exist and even to be unique under certain conditions in the general theory. To demonstrate that the algorithm of ValEncIA-IVP works in the discontinuous case, we need to show that a certain fixed point theorem can be applied to the operator

$$A(x)(t) := x_0 + \int_0^t f(x(s))ds$$

and to the problem $\boldsymbol{x}(t) = A(\boldsymbol{x}(t))$ to also compute the enclosure of the true solution of the IVP for discontinuous right sides f. We exclude sliding solutions for now. The algorithm proposed below would work in this case too, but with a large overestimation.

ValEncIA-IVP uses an a posteriori approach to enclose the solution of the IVP (14) with a continuously differentiable right side f. The true solution $x^*(t)$ is enclosed by the tube $\boldsymbol{x}(t)$ consisting of a non-verified approximation $\tilde{x}(t)$ and a verified error bound $\boldsymbol{R}(t)$ according to

$$x^*(t) \in \boldsymbol{x}(t) := \tilde{x}(t) + \boldsymbol{R}(t). \tag{17}$$

The basic version of the ValEncIA-IVP algorithm for the time interval $[0, T]$ is shown in Figure 1. To improve the tightness of the enclosure, the integration interval $[0, T]$ can be subdivided into smaller intervals. The details about the algorithm, its implementation and the proof of the verified nature of its results in the smooth case are to be found in [5, 32].

To apply this algorithm also for non-smooth functions, we have to use the appropriate definition of the derivative from [2] in Line 3. Since the mean value theorem holds for the generalized derivative definitions, i.e. $f(x) \in f(x_0) + f'(\boldsymbol{x})(\boldsymbol{x} - x_0)$ for all $x, x_0 \in \boldsymbol{x}$, the exchange of derivative definitions is valid. Next, we need to prove that the fixed point iteration in Line 6 is true for this class of functions. In particular, we have to find a fixed point theorem which can be applied in this case. If the

```
Start with $\boldsymbol{x}^{(0)}$, $\tilde{x}(t)$, $\boldsymbol{R}(0)$
For $k = 1 \dots k_{max}$ or while $\dot{\boldsymbol{R}}^{(k)}([0,T])!=\dot{\boldsymbol{R}}^{(k-1)}([0,T])$
  Evaluate $f(\boldsymbol{x}^{(k-1)}) :=$
    $\left( f(\mathrm{m}(\boldsymbol{x}^{(k-1)})) + \sum_{i=1}^{n} \frac{\partial f(\boldsymbol{x}^{(k-1)})}{\partial x_i^{(k-1)}} \cdot \left( \boldsymbol{x}_i^{(k-1)} - \mathrm{m}(\boldsymbol{x}_i^{(k-1)}) \right) \right) \cap f(\boldsymbol{x}^{(k-1)})$
  where $\boldsymbol{x}^{(k-1)} := \boldsymbol{x}^{(k-1)}([0,T])$ and $\mathrm{m}(\boldsymbol{x})$ is the interval midpoint of $\boldsymbol{x}$
  Compute $\dot{\boldsymbol{R}}^{(k)}([0,T]) := -\dot{\tilde{x}} + f(\boldsymbol{x}^{(k-1)})$
  If $\dot{\boldsymbol{R}}^{(k)}([0,T]) \subseteq \dot{\boldsymbol{R}}^{(k-1)}([0,T])$ then
    $\dot{\boldsymbol{R}}^{(k)}([0,T]) := \dot{\boldsymbol{R}}^{(k)}([0,T]) \cap \dot{\boldsymbol{R}}^{(k-1)}([0,T])$
    $\boldsymbol{R}^{(k)}([0,T]) := \boldsymbol{R}(0) + \dot{\boldsymbol{R}}^{(k)}([0,T]) \cdot [0,T]$
    $\boldsymbol{x}^{(k)}([0,T]) := \tilde{x} + \boldsymbol{R}^{(k)}([0,T])$
```

Figure 1: Algorithm of the basic version of ValEncIA-IVP.

right-hand side $f(x)$ of the IVP (14) is Lipschitz continuous for $[0,T]$, then Banach's theorem can be applied as usual to the integral operator $A(\boldsymbol{x})$, which can be shown to be contracting (cf. [41]). Then, the solution exists and is unique. If f is continuous, then Schauder's theorem ensures that the computed enclosure contains a solution to the IVP.

In case of discontinuous functions, Kakutani's fixed point theorem [9] for infinite dimensions (the Fan-Glicksberg theorem) can be applied: If X is a non-empty, compact and convex subset of a locally convex space $\mathbb{E}$ and $\varphi : X \mapsto S(X)$ is a compact and convex set-valued function with a closed graph from X to the set of its non-empty subsets, then φ has a fixed point. Intervals are non-empty, compact and convex. The interval evaluation of a discontinuous right side f according to the definition given in [2] corresponds to a continuous set-valued function with (point) intervals as (convex) sets (that is, it is at least upper semicontinuous). Since integration preserves the continuity properties, the integral operator $A(\boldsymbol{x})$ is also upper semicontinuous and possesses a fixed point if the inclusion property holds.

4.2 Implementation

We implemented the generalized definitions for the interval extension of (15) and its derivative from [2] as a C++ template `pwFunc` allowed to work with a floating point data type (e.g. built-in `double`), an interval data type (e.g. `INTERVAL` from PROFIL/BIAS [14]), and a data type generating first derivatives of smooth functions in the usual sense (e.g. `F<INTERVAL>` from FADBAD++). We chose those software libraries since the solver for smooth IVPs ValEncIA-IVP used them initially. The first situation corresponds to a pointwise evaluation of the function in (15), the second to its interval evaluation and the third one to the generalized evaluation of its derivative.

The dead-time function from the Eq. (16) with $m = h = x_+ = 1$ can be introduced

```
1  template <class T> T phi0(const T& x){ return T(-1.0);}
2  template <class T> T phi1(const T& x){ return T(0.0);}
3  template <class T> T phi2(const T& x){ return T(1.0);}
4  template<class T>  T phi(const T& x_){
5    std::vector<INTERVAL> c;
6    c.push_back(interval(-1));c.push_back(interval(1));
7    std::vector< typename pwFunc<T>::ptrFct> f;
8    f.push_back(&phi0<T>);f.push_back(&phi1<T>);f.push_back(&phi2<T>);
9    pwFunc<T> fp(c, functions);
10   T fx=fp(x_);
11   return fx; }
```

Figure 2: Coding the dead time function from Eq. (16).

into the goal system as shown in Figure 2. In Lines 1-3, each of the smooth functions φ_i, $i = 0, 1, 2$ is defined; in Lines 4-11, the dead time function itself is specified. Here, the vector `c` contains the switching points $c_0 = -h$, $c_1 = h$, the vector `f` the functions φ_i, and the object `fp` provides the necessary evaluations.

That is, for example, if we are interested in solving the following oscillator example from [19], page 45, which models both a sliding pendulum (mechanics) and a relay (electrical circuits),

$$\begin{cases} m\ddot{x} & = & -f(x), \quad \text{where } f \text{ as in (16)} \\ x(0) & = & 2 \\ \dot{x}(0) & = & 0 \end{cases} \tag{18}$$

we just need to define the system function shown in Figure 3 in ValEncIA-IVP and specify the initial values.

4.3 Examples

As an example of the applicability of the method described in previous subsections, we consider several real life motivated systems with and without known exact or reference solutions.

Example 1. The first IVP models the already mentioned oscillator (18), for which

```
template<typename T1, typename T2, typename T3>
T1 system(T1 y, T1 par, T2 x, T3 t_var){
    T1 fvec(SYSTEM_ORDER);
    fvec[0] = y[1];
    fvec[1] = -1.0*phi(y[0]);
    return fvec;
}
```

Figure 3: The goal system code in ValEncIA-IVP for the oscillator (18).

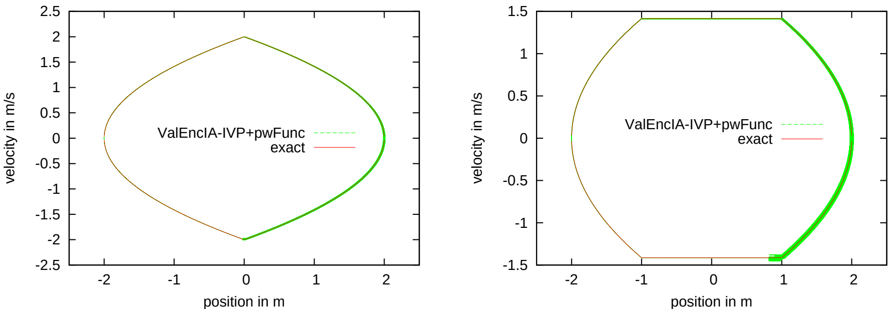

Figure 4: The bounds for the oscillator (18) with (right) and without (left) dead time along with the corresponding exact solutions (Example 1).

the right side f is defined either as in (16) (Situation 2) or as

$$f_1(x_1) = \begin{cases} -h, & x < 0 \\ +h, & x > 0 \end{cases} \tag{19}$$

(Situation 1). For the latter situation, the correspondence between x and $\dot{x}$ is

$$\dot{x} = \pm\sqrt{2(x_0 - x)} \ , \tag{20}$$

if $m = h = 1$, $x_0 = 2$. That is, we obtain parabolas with their apex on the x axis (cf. Figure 4, left, the red curve). Situation 2 for $x_+ = 1$ is almost the same, only that the parabolas stop at $x = \pm x_+$ and are connected with straight lines (Figure 4, right, the red curve). The green curves in the figure denote the results of respective simulations. The simulated solution contains the exact one in both cases. The enclosures start getting wider after the first full cycle.

Example 2. The next problem with the known exact solution is a simplified model for the collapse of the Tacoma Narrows suspension bridge [42]:

$$\begin{cases} \dot{x}_1 & = & x_2 \\ \dot{x}_2 & = & \frac{1}{m}\left(\sin(4t) - q(x_1)\right) \\ x_1(0) & = & 0 \\ x_2(0) & = & 1, \end{cases}$$

where $\sin(4t)$ models the applied force and $q(x_1)$ the upward/downward restoring force

$$q(x_1) = \begin{cases} x_1, & x_1 < 0 \\ 4x_1, & x_1 > 0 \ . \end{cases}$$

The exact solution for $m = 1$ in the first cycle which the model makes in the time $t \in [0, \frac{3\pi}{2}]$ is

$$x_1(t) = \begin{cases} \sin(2t) \cdot \left(\frac{1}{2}(1 + \frac{1}{3}) - \frac{1}{6}\cos(2t)\right), & 0 \le t \le \frac{\pi}{2} \\ \cos t \cdot \left((1 + \frac{2}{5}) - \frac{4}{15} \cdot \sin t \cos(2t)\right), & \frac{\pi}{2} \le t \le \frac{3\pi}{2}. \end{cases}$$

In Figure 5, left, the exact and the simulated solution are shown. Again, the exact solution lies inside the verified bounds. The bounds get wider with time.

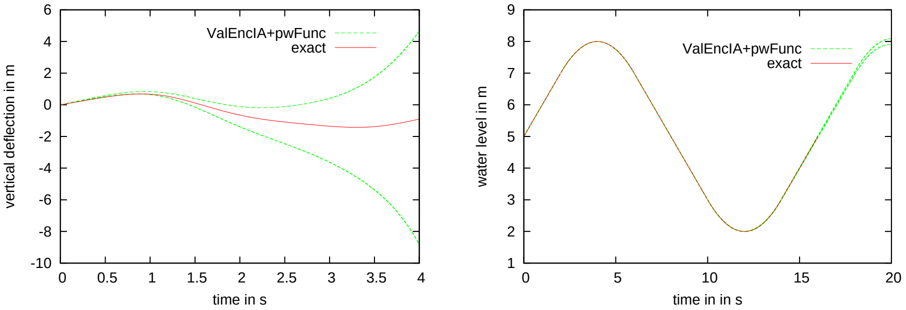

Figure 5: The bounds on the deflection of the suspension bridge (left) and on the water level (right) along with the corresponding exact solutions in the first cycle (Examples 2 and 3).

Example 3. The next example is the model of water level from [26]:

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= 0.5u(x_1) \\ x_1(0) &= 5 \\ x_2(0) &= 1, \end{cases} \qquad u(x_1) = \begin{cases} 1, & x_1 < 3 \\ -1, & x_1 > 7 \\ 0, & \text{otherwise.} \end{cases}$$

The simulation results are shown in Figure 5, right. The exact solution in the first full cycle for $t \in [0, 16]$ is shown in red and correspond to

$$x(t) = \begin{cases} t+5 & \text{for } t \in [0,2], & -\frac{t^2}{4} + 2t + 4 & \text{for } t \in [2,6], \\ -t+13 & \text{for } t \in [6,10], & \frac{t^2}{4} - 6t + 38 & \text{for } t \in [10,14], \\ t-11 & \text{for } t \in [14,18] \ . \end{cases}$$

The width of the enclosure at $t = 35$ is $\text{wid}(\boldsymbol{x}_1) = 0.28$ as opposed to $\text{wid}(\boldsymbol{x}_1) = 10^{-7}$ reported in [26] for their method. Note that the solver from [26] provides the tightened enclosure at the point t_j, whereas our method computes the enclosure comparable to the rough enclosure obtained with Picard's iteration over the interval $[t_{j-1}, t_j]$. In doing this, it produces bounds comparable with the a priori bounds computed in [26], which are much larger (and are subsequently reduced with the help of more advanced techniques, e.g. the Taylor expansion method at a certain point). There are no data about computing times in [26]. However, we expect our technique to be faster, since we do not need to solve systems of algebraic equations to find enclosures of switching points.

Example 4. This example has no classical solution after $t_* \approx 2.03$ [10, 26]:

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{1}{5}x_2 - x_1 + 2\cos(\pi t) - u(x_2) \\ x_1(0) &= 3 \\ x_2(0) &= 4 \end{cases} \qquad u(x_2) = \begin{cases} -4, & x_2 < 0 \\ +4, & x_2 > 0 \ . \end{cases} \tag{21}$$

After t_*, the velocity x_2 becomes zero until $t_\star \approx 2.6$, where the solution leaves the switching surface $x_2 = 0$. In this case, a solution definition on the switching surface

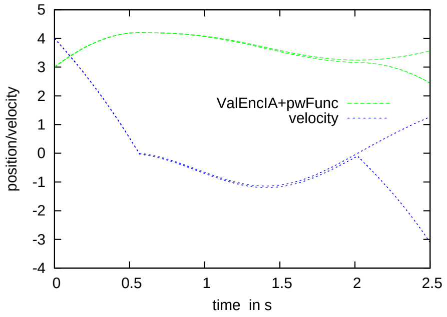

Figure 6: The bounds on the position and velocity of (21) obtained with the extended version of ValEncIA-IVP (Example 4).

is necessary, for example, in the Filippov sense (8). In [26], the authors deal with the problem by breaking of the solution process after t_*. The algorithm in [35] is reported to produce a solution according to Eq. (8) between t_* and $t_\star$. However, the authors do not solve exactly the same problem there (but a similar one without $2\cos(\pi t)$ on the right side and with ±1 instead of ±4), so a direct comparison of results is not possible. The simulation results of our method are shown in Figure 6 for the position x_1 and the velocity x_2. The theory presented here has no means of determining whether the solution stays on the switching surface. According to the definition given in [2], a convex hull of both possible branches has to be constructed, leading eventually to a large overestimation of the true solution set.

Two more close to life examples are the mechanical system with friction and hysteresis from [2] and sliding control of a technical system from [5]. They demonstrate that our method can be used in practice as an easy means of the first analysis in spite of the overestimation observed in the previous examples.

Example 5. The model for a system with friction and hysteresis we consider was introduced in [33] and is represented according to the graph in Figure 7. The reformulation of this system for the extended version of ValEncia-IVP presented here is

$$\left\{\begin{array}{lcl} \dot{\boldsymbol{x}}_1 & = & \boldsymbol{x}_2 \\ \dot{\boldsymbol{x}}_2 & = & \frac{1}{\boldsymbol{m}}(2\sin(3\boldsymbol{x}_4) - \kappa_x \boldsymbol{x}_1 - \kappa_\omega \boldsymbol{x}_3 - \varphi_1(\boldsymbol{x}_2)) \\ \dot{\boldsymbol{x}}_3 & = & \rho \cdot \big(\boldsymbol{x}_2 - \sigma \cdot \varphi_2(\boldsymbol{x}_2) \cdot \varphi_2(\boldsymbol{x}_3)^{\nu-1} \cdot \boldsymbol{x}_3 \\ & & + (\sigma - 1) \cdot \boldsymbol{x}_2 \cdot \varphi_2(\boldsymbol{x}_3)^{\nu}\big) \\ \dot{\boldsymbol{x}}_4 & = & 1, \end{array}\right. \tag{22}$$

where φ_1 is the friction force (including static and sliding parts) and φ_2 is the absolute value function specified in terms of the definition in Eq. (15) as

$$\varphi_1(x_2) = \left\{\begin{array}{rl} -F_s + \mu \cdot x_2, & x_2 < 0 \\ F_s + \mu \cdot x_2, & x_2 > 0 \end{array}\right. \tag{23}$$

$$\varphi_2(x_i) = |x_i| = \left\{\begin{array}{rl} -x_i, & x_i < 0 \\ x_i, & x_i > 0 \end{array}\right. \tag{24}$$

for an interval state vector $\boldsymbol{x} = (\boldsymbol{x}_1\ \boldsymbol{x}_2\ \boldsymbol{x}_3\ \boldsymbol{x}_4)^{\mathrm{T}}$, $x_i \in \boldsymbol{x}_i$, $i = 2, 3$ and the static friction coefficient $F_s \in \boldsymbol{F}_s$.

The authors in [33] solved this problem by applying a Taylor series verified enclosure method to the transition graph (cf. Section 2). In this application, the transition conditions T_i^j describe those state and control-input-dependent relations which lead to the activation of the discrete model state S_j from the currently active state S_i, as shown in Figure 7.

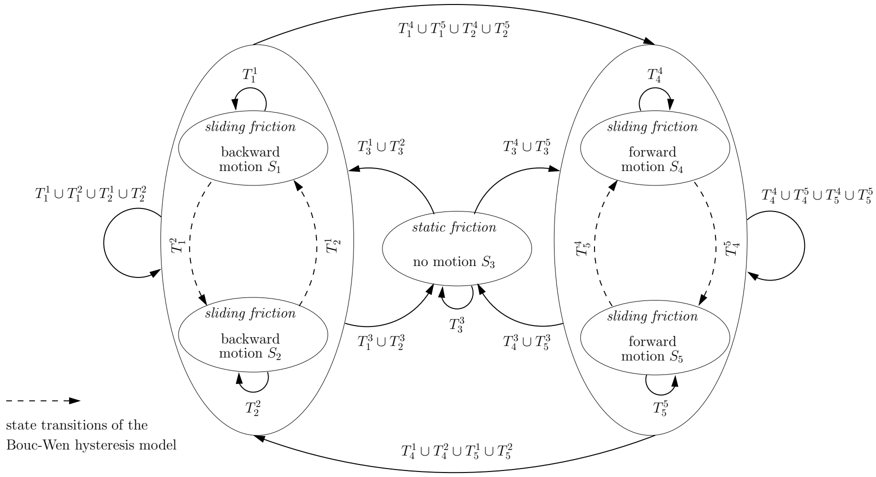

Figure 7: State transition diagram for all switchings between discrete model states for the system with friction and hysteresis [33] (Example 5).

Our bounds for the velocity $\boldsymbol{x}_2$ of the mass for the non-degenerate case (where the solution does not stay at the switching surface) is shown in the Figure 8, left. The values of parameters we used are given by Table 2. We compared the results to those obtained with the MATLAB simulation from [33]. As expected, the figures demonstrate that the results produced by the approach proposed in this paper are consistent with those from [33] (the interval widths are almost the same). The simulation in VALENCIA-IVP with step size 0.01 takes approximately 3 seconds CPU time on an Intel Xeon 2GHz multicore processor under Linux 2.6.23.14-115.fc8. The MATLAB[3] simulation needs 376 seconds on a Intel Core2 Duo E8400 3GHz computer under Windows 7 Professional. Another advantage of our approach is that we do not need to track down all possible transitions T_i^j manually as shown in Figure 7. The equations (22)–(24) suffice for describing the system.

Example 6. The second application of our method is for controlling the temperature of a stack of solid oxide fuel cells (SOFC). A simple one-dimensional model for the temperature of the SOFC stack was described in [5]. Since this IVP is too long to repeat it here in full, we summarize its structure only as the following equation:

$$\vartheta_{FC} = a(\vartheta_{FC}, p, d) + b(\vartheta_{FC}, p, d) * v(t), \qquad (25)$$

where a and b are polynomials in the temperature ϑ_{FC}, $p \in \boldsymbol{p}$ are parameters, $d \in \boldsymbol{d}$ is the possible disturbance, and $v(t)$ accounts for the control law. The goal is to heat the system until a certain maximum temperature is reached ($\vartheta_{stable} = 880$K in our case),

[3] MATLAB R2011a (64 bit) with INTLAB V6

Table 2: Parameter values for the mechanical system with friction and hysteresis from [33] (Example 5).

Parameter	Value	Parameter	Value
$\boldsymbol{x}_1(0)$	0 m	$\boldsymbol{x}_3(0)$	-0.001
κ_ω	0.001	$\boldsymbol{m}$	[1.1,1.21] kg
$\boldsymbol{F}_s$	[0.15,0.03] N	σ	0.001
ρ	0.001	μ	0.001
ν	1	$u(t)$	$2\sin(3t)$ N
$\boldsymbol{x}_2(0)$	0 m/s	κ_x	0.001

and hold that temperature afterwards, taking into account possible disturbances and uncertainties in parameters. How to devise the function $v(t)$ in such a way is described in detail in [5]. Here, we use the system (25) with the following control law

$$v = \begin{cases} v_1 & \text{for } \vartheta_{FC} < 800\text{K}, \\ v_2 & \text{for } 800\text{K} < \vartheta_{FC} < 880\text{K}, \\ v_3 & \text{for } \vartheta_{FC} > 880\text{K}, \end{cases} \tag{26}$$

where $v_1 := \dot{m}_a(\vartheta_{FC} - \vartheta_{CG})$,

$$v_2 := -\frac{a\left(\vartheta_{FC}(t), p, d\right)}{b\left(\vartheta_{FC}(t), p, d\right)} - \frac{-\tilde{\eta}}{b\left(\vartheta_{FC}(t), p, d\right)} \quad , \quad v_3 := -\frac{a\left(\vartheta_{FC}(t), p, d\right)}{b\left(\vartheta_{FC}(t), p, d\right)} - \frac{\tilde{\eta}}{b\left(\vartheta_{FC}(t), p, d\right)}$$

with $\tilde{\eta} = 0.1$, $\dot{m}_a$, ϑ_{CG} constants in this setting. The point interval parameters p are assumed to be constant. We consider the disturbance $d(t) = 0.1\text{sign}(\sin(0.001t))$ bounded by the interval $[-0.1, 0.1]$ with the initial condition $\vartheta_{FC}(0) = 790\text{K}$. This is a realistic situation for which the switching points are isolated. The results of the verified simulation of this system are shown in Figure 8 on the right. The changes in the control law at $\vartheta_{FC} = 800\text{K}$ and $\vartheta_{FC} = 880\text{K}$ are taken into account appropriately.

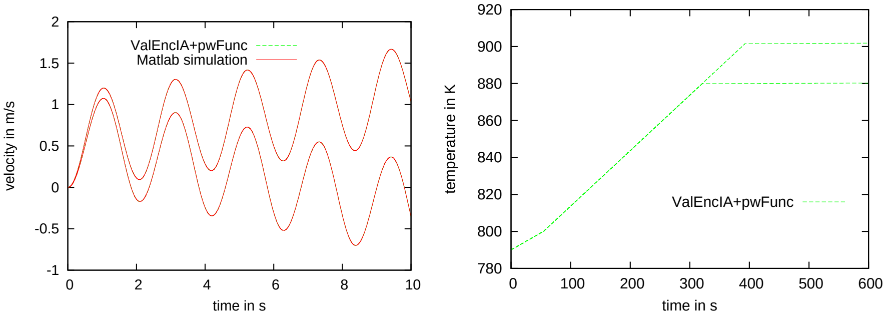

Figure 8: Velocity of the mass for the system in Eq. (22) compared to the automaton-based method from [33] (left) and the bounds for the temperature of the SOFC stack in Eq. (25) on the right (Examples 5 and 6).

5 Conclusions and Outlook

We presented an overview of the current state of the art in the area of modeling and verified simulation of non-smooth systems. This area can be observed to be somewhat underdeveloped in comparison to verified simulation of smooth IVPs. In particular, it is difficult to find a modern, flexible, ready-to-use verified software package for general tasks[4]. This was the reason we developed a simple means for the analysis of IVPs with non-smooth right sides based on the smooth theory and a generalization of the evaluation and derivation notions for functions with discontinuities or non-differentiable ones. Assumptions for the use of the approach are the isolated nature of the switching points and the possibility to represent the right side as a chain of non-smooth functions depending on one variable. The approach was implemented in C++ and applied in combination with the basic version of the smooth solver ValEncIA-IVP to simulate close to life models of a mechanical system with friction and hysteresis as well as of controlled temperature of a fuel cell stack.

An extensive comparison with the literature on existing verified methods[5] and with exact solutions revealed that although the results produced by our method are verified, the computed bounds are overconservative, and overestimate the true solution set. One reason for this is that we use only the basic version of ValEncIA-IVP, which provides relatively wide bounds comparable with auxiliary a priori bounds used by methods such as those reported in [26, 35]. An advantage of our approach is the simple and natural way in which the goal system can be introduced into the IVP solver (cf. Section 4.2). We also expect better CPU times (which could not be verified due to the lack of the actual software code or differing programming environments). That is, the potential of the approach presented in this paper lies in the real life applications for which verified bounds over short time intervals can be combined with an otherwise floating point based simulation to deal with, for example, uncertainty.

Our future work will concern, on the one hand, computing tighter enclosures in points and, on the other hand, extending the class of functions to which the proposed generalization of the derivative definition can be applied.

References

[1] V. Acary and B. Brogliato. *Numerical Methods for Nonsmooth Dynamical Systems: Applications in Mechanics and Electronics.* Lecture Notes in Applied and Computational Mechanics. Springer, 2008.

[2] E. Auer, S. Kiel, and A. Rauh. A Verified Method for Solving Piecewise Smooth Initial Value Problems. *International Journal of Applied Mathematics and Computer Science*, 23(4), 2013.

[3] M. Bernardo, C. Budd, A.R. Champneys, and P. Kowalczyk. *Piecewise-smooth Dynamical Systems: Theory and Applications.* Applied Mathematical Sciences. Springer, 2007.

[4] M. Berz and K. Makino. Suppression of the Wrapping Effect by Taylor Model-based Verified Integrators: Long-term Stabilization by Shrink Wrapping. *International Journal of Differential Equations and Applications*, 10(4):385–403, 2005.

[4] The distribution of HyTech [11] to be found on the web seems to allow only for linear automata and requires a reformulation of the problem in this form.

[5] Actual simulation data was available to us only for the automaton based method from [33]

[5] T. Dötschel, E. Auer, A. Rauh, and H. Aschemann. Thermal Behavior of High-Temperature Fuel Cells: Reliable Parameter Identification and Interval-Based Sliding Mode Control. *Soft Computing*, 17(8):1329–1343, 2013.

[6] I. Eble. *Über Taylor-Modelle.* PhD thesis, University of Karlsruhe, 2007. In German.

[7] A. Eggers, M. Fränzle, and C. Herde. Application of Constraint Solving and ODE-Enclosure Methods to the Analysis of Hybrid Systems. In *Numerical Analysis and Applied Mathematics 2009*, volume 1168, pages 1326–1330. American Institute of Physics, 2009.

[8] A. Filippov. *Differential Equations with Discontinuous Righthand Sides.* Kluwer Academic Publishers, 1988.

[9] A. Granas and J. Dugundji. *Fixed Point Theory.* Springer Monographs in Mathematics. Springer, 2003.

[10] E. Hairer, S. P. N*o*rsett, and G. Wanner. *Solving Ordinary Differential Equations I (2nd revised. ed.): Nonstiff Problems.* Springer, 2009.

[11] Thomas A. Henzinger, Benjamin Horowitz, Rupak Majumdar, and Howard Wong-Toi. Beyond HYTECH: Hybrid Systems Analysis Using Interval Numerical Methods. In *Proceedings of the Third International Workshop on Hybrid Systems: Computation and Control*, HSCC'00, pages 130–144, London, UK, 2000. Springer.

[12] D. Ishii. *Simulation and Verification of Hybrid Systems Based on Interval Analysis and Constraint Programming.* PhD thesis, Waseda University, Japan, 2010.

[13] R. B. Kearfott. *Rigorous Global Search: Continuous Problems.* Kluwer, 1996.

[14] O. Knüppel. PROFIL/BIAS — A Fast Interval Library. *Computing*, 53(3):277–287, 1994.

[15] Markus Kunze. *Non-Smooth Dynamical Systems.* Springer, 2000.

[16] R. Lohner. *Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen.* PhD thesis, Universität Karlsruhe, 1988. In German.

[17] R. Lohner. Computation of guaranteed solutions of ordinary initial and boundary value problems. In J. R. Cash and I. Gladwell, editors, *Computational Ordinary Differential Equations*, pages 425–435, Oxford, UK, 1992. Clarendon Press.

[18] J. Lunze and F. Lamnabhi-Lagarrigue. *Handbook of Hybrid Systems Control – Theory, Tools, Applications.* Cambridge University Press, Cambridge, UK, 2009.

[19] K. Magnus and K. Popp. *Schwingungen.* Leitfäden der angewandten Mathematik und Mechanik. Teubner, 2005. In German.

[20] S. Mahmoud and X. Chen. A Verified Inexact Implicit Runge-Kutta Method for Nonsmooth ODEs. *Numerical Algorithms*, 47:275–290, 2008.

[21] K. Makino and M. Berz. Suppression of the Wrapping Effect by Taylor Model-based Verified Integrators: Long-term Stabilization by Preconditioning. *International Journal of Differential Equations and Applications*, 10(4):353–384, 2005.

[22] K. Makino and M. Berz. Suppression of the Wrapping Effect by Taylor Model-based Verified Integrators: The Single Step. *International Journal of Pure and Applied Mathematics*, 36(2):175–197, 2006.

[23] R. Mannshardt. One-Step Methods of Any Order for Ordinary Differential Equations with Discontinuous Right-Hand Sides. *Numerische Mathematik*, 31:131–152, 1978.

[24] R.E. Moore. *Interval Arithmetic.* Prentice-Hall, New Jersey, 1966.

[25] H. Munoz and R. B. Kearfott. Slope Intervals, Generalized Gradients, Semigradients, Slant Derivatives, and Csets. *Reliable Computing*, 10:163–193, 2004.

[26] N. Nedialkov and M. von Mohrenschildt. Rigorous Simulation of Hybrid Dynamic Systems with Symbolic and Interval Methods. In *Proceedings of the American Control Conference Anchorage*, 2002.

[27] N. S. Nedialkov. *The Design and Implementation of an Object-Oriented Validated ODE Solver.* Kluwer Academic Publishers, 2002.

[28] N. S. Nedialkov and K. R. Jackson. A New Perspective on the Wrapping Effect in Interval Methods for Initial Value Problems for Ordinary Differential Equations. In *Perspectives on Enclosure Methods*, pages 219–264, Vienna, Austria, 2001. Springer-Verlag.

[29] M. Neher, K. R. Jackson, and N. S. Nedialkov. On Taylor Model Based Integration of ODEs. *SIAM J. Numer. Anal*, 45:236–262, 2007.

[30] Y. Orlov. Finite Time Stability and Robust Control Synthesis of Uncertain Switched Systems. *SIAM J. Control Optim.*, 43(4):1253–1271, April 2004.

[31] S. Ratschan. An Algorithm for Formal Safety Verification of Complex Heterogeneous Systems. In *Proceeding of REC 2012*, pages 457–468, 2012.

[32] A. Rauh and E. Auer. Verified Simulation of ODEs and DAEs in ValEncIA-IVP. *Reliable Computing*, 5(4):370–381, 2011.

[33] A. Rauh, Ch. Siebert, and H. Aschemann. Verified Simulation and Optimization of Dynamic Systems with Friction and Hysteresis. In *Proceedings of ENOC 2011*, Rome, Italy, 2011.

[34] R. Rihm. Enclosing Solutions with Switching Points in Ordinary Differential Equations. In *Computer arithmetic and enclosure methods. Proceedings of SCAN 91*, pages 419–425. Amsterdam: North-Holland, 1992.

[35] R. Rihm. *Über Einschließungsverfahren für gewöhnliche Anfangswertprobleme und ihre Anwendung auf Differentialgleichungen mit unstetiger rechter Seite.* PhD thesis, Universität Karlsruhe, 1993. In German.

[36] R. Rihm. Implicit Methods for Enclosing Solutions of ODEs. *"Journal of Universal Computer Science"*, 4(2):202–209, 1998.

[37] M. Schnurr and D. Ratz. Slope Enclosures for Functions Given by Two or More Branches. *BIT Numerical Mathematics*, 48(4):783–797, 2008.

[38] O. Stauning. *Automatic Validation of Numerical Solutions.* PhD thesis, Technical University of Denmark, Lyngby, 1997.

[39] D. Stewart. A High Accuracy Method for Solving ODEs with Discontinuous Right-Hand Side. *Numerische Mathematik*, 58:299–328, 1990.

[40] D. Strobach, A. Kecskeméthy, G. Steinwender, and E. B. Zwick. Rapid identification of muscle activation profiles via optimization and smooth profile patches. *Materialwissenschaft und Werkstofftechnik*, 36(12):802–813, 2005.

[41] W. Walter. *Ordinary Differential Equations.* Number 182 in Graduate Texts in Mathematics. Springer, 1998.

[42] D.G. Zill and M.R. Cullen. *A First Course in Differential Equations with Modeling Applications.* Brooks/Cole, 2001.