15-424: Foundations of Cyber-Physical Systems

Lecture Notes on Differential Equations & Domains

André Platzer

Carnegie Mellon University

Lecture 2

1. Introduction

In the last lecture, we have learned about the characteristic features of *cyber-physical systems* (CPS): they combine cyber capabilities (computation and/or communication) with physical capabilities (motion or other physical processes). Cars, aircraft, and robots are prime examples, because they move physically in space in a way that is determined by discrete computerized control algorithms. Designing these algorithms to control CPSs is challenging due to their tight coupling with physical behavior. At the same time, it is vital that these algorithms be correct, since we rely on CPSs for safety-critical tasks like keeping aircraft from colliding.

Since CPS combine cyber and physical capabilities, we need to understand both to understand CPS. It is not enough to understand both in isolation, though, because we also need to understand how the cyber and the physics work together, i.e. what happens when they interface and interact, because this is what CPSs are all about.

You already have experience with models of computation and algorithms for the cyber part of CPS, because you have seen the use of programming languages for computer programming in previous courses. In CPS, we do not program computers, but program CPS instead. So we program computers that interact with physics to achieve their goals. In this lecture, we study models of physics and the most elementary part of how they can interact with cyber. Physics by and large is obviously a deep subject. But for CPS one of the most fundamental models of physics is sufficient, that of ordinary differential equations.

While this lecture covers the most important parts of differential equations, it is not to be understood as doing complete diligence to the area of ordinary differential equations. You are advised to refer back to your differential equations course and follow the supplementary information[1] available on the course web page as needed during this course. We refer to the book by Walter [Wal98] for details and proofs about differential equations. For further background on differential equations, we refer you to the literature [Har64, Rei71, EEHJ96].

These lecture notes are based on material on cyber-physical systems, hybrid programs, and logic [Pla12, Pla10, Pla08, Pla07]. Cyber-physical systems play an important role in numerous domains [PCA07, LS10, LSC+12] with applications in cars [DGV96], aircraft [TPS98], robots [PKV09], and power plants [FKV04], chemical processes [RKR10, KGDB10], medical models [GBF+11, KAS+11], and even an importance for understanding biological systems [Tiw11].

More information about CPS can be found in [Pla10, Chapter 1]. Differential equations and domains are described in [Pla10, Chapter 2.2,2.3] in more detail.

2. Differential Equations as Models of Continuous Physical Processes

Differential equations model processes in which the (state) variables of a system evolve continuously in time. A differential equation concisely describes how the system evolves over time. It describes how the variables change locally, so it, basically, indicates the direction in which the variables evolve at each point in space. Fig. 1 shows the respective directions in which the system evolves by a vector at each point and illustrates one solution which follows those vectors everywhere. Of course, the figure would be rather cluttered if we would literally try to indicate the vector at each and every point, of which there are uncountably infinitely many. But this is a shortcoming only of our illustration. Differential equations actually define such a vector for the direction of evolution at every point in space.

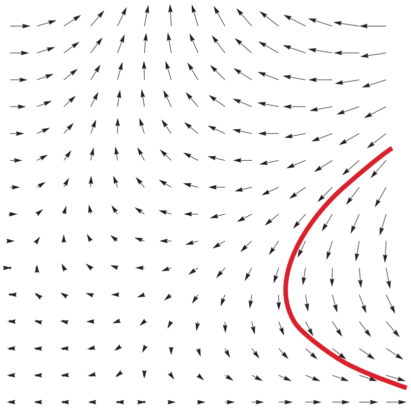

Figure 1: Vector field and one solution of a differential equation

[1] http://symbolaris.com/course/fcps13-resources.html

As an example, suppose we have a car whose position is denoted by x. How the value of variable x changes over time depends on how fast the car is driving. Let v denote the velocity of the car. Since v is the velocity of the car, its position x changes such that its derivative x' is v, which we write by the differential equation $x' = v$. This differential equation is supposed to mean that the time-derivative of the position x is the velocity v. So how x evolves depends on v. If the velocity is $v = 0$, then the position x does not change at all. If $v > 0$, then the position x keeps on increasing. How fast x increases depends on the value of v, bigger v give quicker changes in x.

Of course, the velocity v, itself, may also be subject to change over time. The car might accelerate, so let a denote its acceleration. Then the velocity v changes with time-derivative a, so $v' = a$. Overall, the car then follows the differential equation (system):[2]

$$x' = v, v' = a$$

That is, the position x of the car changes with time-derivative v, which, in turn, changes with time-derivative a.

What we mean by this differential equation, intuitively, is that the system has a vector field where all vectors point into direction a. What does this mean exactly?

3. The Meaning of Differential Equations

We relate some intuitive concept to how differential equations describe the direction of the evolution of a system as a vector field Fig. 1. But what exactly is a vector field? What does it mean to describe directions of evolutions at every point in space? Could these directions not possibly contradict each other so that the description becomes ambiguous? What is the exact meaning of a differential equation in the first place?

The only way to truly understand any system is to understand exactly what each of its pieces does. CPSs are demanding and misunderstandings about their effect often have far-reaching consequences. The physical impacts of CPSs do not leave much room for failure, so we immediately want to get into the mood of consistently studying the behavior and exact meaning of all relevant aspects of CPS.

An ordinary differential equation in explicit form is an equation $y'(t) = f(t, y)$ where $y'(t)$ is meant to be the derivative of y with respect to time t. A solution is a differentiable function Y which satisfies this equation when substituted in the differential equation, i.e., when substituting $Y(t)$ for y and the derivative $Y'(t)$ of Y at t for $y'(t)$.

Definition 1 (Ordinary differential equation)**.** Let $f : D \to \mathbb{R}^n$ be a function on a domain $D \subseteq \mathbb{R} \times \mathbb{R}^n$. The function $Y : I \to \mathbb{R}^n$ is a *solution* on the interval $I \subseteq \mathbb{R}$ of the *initial value problem*

$$\begin{bmatrix} y'(t) = & f(t, y) \\ y(t_0) = & y_0 \end{bmatrix} \tag{1}$$

[2] Note that the value of x changes over time, so it is really a function of time. Hence, the notation $x'(t) = v(t), v'(t) = a$ is sometimes used. It is customary, however, to suppress the argument t for time and just write $x' = v, v' = a$ instead.

with *ordinary differential equation* (*ODE*) $y' = f(t, y)$, if, for all $t \in I$

1. $(t, Y(t)) \in D$,
2. $Y'(t)$ exists and $Y'(t) = f(t, Y(t))$,
3. $Y(t_0) = y_0$.

If $f : D \to \mathbb{R}^n$ is continuous, then it is easy to see that $Y : I \to \mathbb{R}^n$ is continuously differentiable. Similarly if f is k-times continuously differentiable then Y is $k+1$-times continuously differentiable. The definition is accordingly for higher-order differential equations, i.e., differential equations involving higher-order derivatives $y^{(n)}(t)$ for $n > 1$.

Let us consider the intuition for this definition. A differential equation (system) can be thought of as a vector field such as the one in Fig. 1, where, at each point, the vector shows in which direction the solution evolves. At every point, the vector would correspond to the right-hand side of the differential equation. A solution of a differential equation adheres to this vector field at every point, i.e., the solution (e.g., the solid line in Fig. 1) locally follows the direction indicated by the vector of the right-hand side of the differential equation. There are many solutions of the differential equation corresponding to the vector field illustrated in Fig. 1. For the particular initial value problem, however, a solution also has to start at the position y_0 at time t_0 and then follow the differential equations or vector field from this point. In general, there could still be multiple solutions for the same initial value problem.

Example 2. Some differential equations are easy to solve. The initial value problem

$$\begin{bmatrix} x'(t) = & 5 \\ x(0) = & 2 \end{bmatrix}$$

has a solution $x(t) = 5t + 2$. This can be checked easily by inserting the solution into the differential equation and initial value equation:

$$\begin{bmatrix} (x(t))' = & (5t+2)' = 5 \\ x(0) = & 5 \cdot 0 + 2 = 2 \end{bmatrix}$$

Example 3. Consider the initial value problem

$$\begin{bmatrix} x'(t) = & -2x \\ x(1) = & 3 \end{bmatrix}$$

which has a solution $x(t) = 3e^{-2(t-1)}$. The test, again, is to insert the solution into the (differential) equations of the initial value problems and check:

$$\begin{bmatrix} (3e^{-2(t-1)})' = & -6e^{-2(t-1)} = -2x(t) \\ x(1) = & 3e^{-2(1-1)} = 3 \end{bmatrix}$$

Example 4. Consider the differential equation system $z' = v, v' = a$ and the initial value problem

$$\begin{bmatrix} z'(t) = & v(t) \\ v'(t) = & a \\ z(0) = & z_0 \\ v(0) = & v_0 \end{bmatrix}$$

Note that this initial value problem is a *symbolic initial value problem* with symbols z_0, v_0 as initial values (not specific numbers like 5 and 2.3). Moreover, the differential equation has a constant symbol a, and not a specific number like 0.6, in the differential equation. In vectorial notation, the initial value problem with this differential equation system corresponds to a vectorial system when we denote $y(t) := (z(t), v(t))$, i.e., with dimension $n = 2$ in Def. 1:

$$\begin{bmatrix} y'(t) = \begin{pmatrix} z \\ v \end{pmatrix}'(t) = & \begin{pmatrix} v(t) \\ a \end{pmatrix} \\ y(0) = \begin{pmatrix} z \\ v \end{pmatrix}(0) = & \begin{pmatrix} z_0 \\ v_0 \end{pmatrix} \end{bmatrix}$$

The solution of this initial value problem is

$$\begin{aligned} z(t) &= \frac{a}{2}t^2 + v_0 t + z_0 \\ v(t) &= at + v_0 \end{aligned}$$

We can show that this is the solution by inserting the solution into the (differential) equations of the initial value problems and checking:

$$\begin{bmatrix} (\frac{a}{2}t^2 + v_0 t + z_0)' = & 2\frac{a}{2}t + v_0 = v(t) \\ (at + v_0)' = & a \\ z(0) = & \frac{a}{2}0^2 + v_0 0 + z_0 = z_0 \\ v(0) = & a0 + v_0 = v_0 \end{bmatrix}$$

Example 5. Consider the differential equation system $x' = y, y' = -x$ and the initial value problem

$$\begin{bmatrix} x'(t) = & y(t) \\ y'(t) = & -x(t) \\ x(0) = & 1 \\ y(0) = & 1 \end{bmatrix}$$

The solution of this initial value problem is

$$\begin{aligned} x(t) &= \cos(t) + \sin(t) \\ y(t) &= \cos(t) - \sin(t) \end{aligned}$$

We can show that this is the solution by inserting the solution into the (differential) equations of the initial value problems and checking:

$$\left[\begin{array}{rl} (\cos(t)+\sin(t))' = & -\sin(t)+\cos(t) = y(t) \\ (\cos(t)-\sin(t))' = & -\sin(t)-\cos(t) = -x(t) \\ x(0) = & \cos(0)+\sin(0) = 1 \\ y(0) = & \cos(0)-\sin(0) = 1 \end{array}\right]$$

Note 1 (Descriptive power of differential equations)**.** *As a general phenomenon, observe that solutions of differential equations can be much more involved than the differential equations themselves, which is part of the representational and descriptive power of differential equations.*

4. Domains of Differential Equations

Now we understand exactly what a differential equation is and how it describes a continuous physical process. In CPS, however, physical processes interact with cyber elements such as computers. When and how do physics and cyber elements interact? The first thing we need to understand for that is how to describe when physics stops so that the cyber elements take control of what happens next. Obviously, physics does not literally stop evolving, but rather keeps on evolving all the time. Yet, the cyber parts only take effect every now and then. So, our intuition may imagine physics “pauses” for a period of duration 0 and lets the cyber take action to influence the inputs that physics is based on.

The cyber and the physics could interface in more than one way. Physics might evolve and the cyber elements interrupt to inspect measurements about the state of the system periodically to decide what to do next. Or the physics might trigger certain conditions that cause cyber elements to compute their responses. Another way to look at that is that a differential equation that a system follows forever without further intervention by anything would not describe a particularly well-controlled system. All those ways have in common that our model of physics needs to come up with information about when it stops evolving to give cyber a chance to perform its task.

This information is what is a called an *evolution domain H* of a differential equation, which describes a region that the system cannot leave. If the system were ever about to leave this region, it would stop evolving right away before it leaves the evolution domain.

Note 2. *A differential equation $x' = f(x)$ with evolution domain H is denoted by*

$$x' = f(x) \,\&\, H$$

This notation $x' = f(x) \,\&\, H$ signifies that the system follows the differential equation $x' = f(x)$ for any duration, but is never allowed to leave the region described by H. So the system evolution has to stop while the state is still in H.

If, e.g., t is a time variable with $t' = 1$, then $x' = v, v' = a, t' = 1 \,\&\, t \le \varepsilon$ describes a system that follows the differential equation at most until time $t = \varepsilon$ and not any further. The evolution domain $H \overset{\text{def}}{\equiv} (v \ge 0)$, instead, restricts the system $x' = v, v' = a \,\&\, v \ge 0$ to nonnegative velocities. Should the velocity ever become negative while following the differential equation $x' = v, v' = a$, then the system stops before that happens.

In the scenario illustrated in Fig. 2, the system starts at time 0 inside the evolution domain that is depicted as a shaded green region in Fig. 2. Then the system follows the differential equation $x' = f(x)$ for any period of time, but has to stop before it leaves H. Here, it stops at time r.

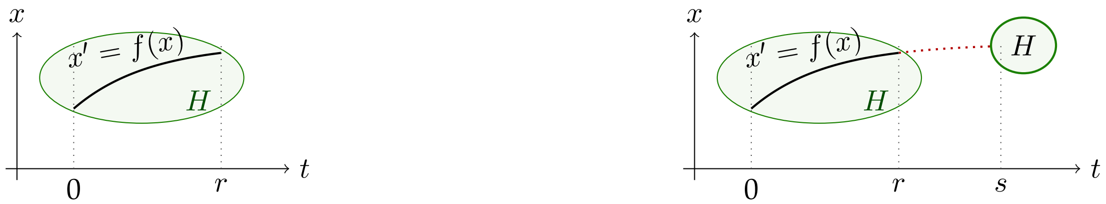

Figure 2: System $x' = f(x) \,\&\, H$ follows the differential equation $x' = f(x)$ but cannot leave the (shaded) evolution domain H.

In contrast, consider the scenario shown on the right of Fig. 2. The system is *not* allowed to evolve until time s, because—even if the system is back in the evolution domain H at that time—it has left the evolution domain H between time r and s (indicated by dotted lines), which is not allowed. Consequently, the continuous evolution on the right of Fig. 2 will also stop at time r at the latest.

How can we properly describe the evolution domain H? We will need some logic for that.

5. Continuous Programs: Syntax

After these preparations for understanding differential equations and domains, we start developing a programming language for cyber-physical systems. Ultimately, this programming language of *hybrid programs* will contain more features than just differential equations. But this most crucial feature is what we start with. This course develops this programming language and its understanding and its analysis in layers one after the other.

Continuous Programs. The first element of the syntax of hybrid programs is the following.

Note 3. *Version 1 of* hybrid programs *(HPs) are* continuous programs. *These are defined by the following grammar (α is a HP, x a variable, θ a term possibly containing x, and H a formula of first-order logic of real arithmetic):*

$$\alpha ::= x' = \theta \,\&\, H$$

This means that a hybrid program α consists of a single statement of the form $x' = \theta \,\&\, H$. In later lectures, we will add more statements to hybrid programs, but focus on differential equations for now. The formula H is called *evolution domain constraint* of the *continuous evolution* $x' = \theta \,\&\, H$. Further x is allowed to be a vector of variables and, then, θ is a vector of terms of the same dimension. This corresponds to the case of differential equation systems such as:

$$x' = v, v' = a \,\&\, (v \geq 0 \land v \leq 10)$$

Differential equations are allowed without an evolution domain constraint H as well, for example:

$$x' = y, y' = x + y^2$$

which corresponds to choosing $true$ for H, since the formula $true$ is true everywhere and imposes no condition on the state.

Terms. A rigorous definition of the syntax of hybrid programs also depends on defining what a term θ is and what a formula H of first-order logic of real arithmetic is. A *term* θ is a polynomial term defined by the grammar (where θ, ϑ are terms, x a variable, and c a rational number constant):

$$\theta, \vartheta ::= x \mid c \mid \theta + \vartheta \mid \theta \cdot \vartheta$$

This means that a term θ is either a variable x, or a rational number constant $c \in \mathbb{Q}$, or a sum of terms θ, ϑ, or a product of terms θ, ϑ. Subtraction $\theta - \vartheta$ is another useful case, but it turns out that it is already included, because subtraction can be defined by $\theta + (-1) \cdot \vartheta$.

First-order Formulas. The formulas of first-order logic of real arithmetic are defined as usual in first-order logic, yet using the language of real arithmetic. The formulas of *first-order logic of real arithmetic* are defined by the following grammar (where F, G are formulas of first-order logic of real arithmetic, θ, ϑ are (polynomial) terms, and x a variable):

$$F, G ::= \theta = \vartheta \mid \theta \geq \vartheta \mid \neg F \mid F \land G \mid F \lor G \mid F \rightarrow G \mid F \leftrightarrow G \mid \forall x\, F \mid \exists x\, F$$

The usual abbreviations are allowed, such as $\theta \leq \vartheta$ for $\vartheta \geq \theta$ and $\theta < \vartheta$ for $\neg(\theta \geq \vartheta)$.

6. Continuous Programs: Semantics

> **Note 4** (Syntax vs. Semantics)**.** *Syntax just defines a notation. Its meaning is defined by the semantics.*

Terms. The meaning of a continuous evolution $x' = \theta \,\&\, H$ depends on understanding the meaning of terms θ. A term θ is a syntactic expression. Its value depends on the interpretation of the variables contained in θ. What values those variables have changes depending on the state of the CPS. A *state* ν is a mapping from variables to real numbers. The set of states is denoted $\mathcal{S}$.

Definition 6 (Valuation of terms)**.** The *value of term* θ in state ν is denoted $[\![\theta]\!]_\nu$ and defined by induction on the structure of θ:

$$\begin{aligned}
[\![x]\!]_\nu &= \nu(x) && \text{if } x \text{ is a variable}\\
[\![c]\!]_\nu &= c && \text{if } c \text{ is a rational constant}\\
[\![\theta + \vartheta]\!]_\nu &= [\![\theta]\!]_\nu + [\![\vartheta]\!]_\nu\\
[\![\theta \cdot \vartheta]\!]_\nu &= [\![\theta]\!]_\nu \cdot [\![\vartheta]\!]_\nu
\end{aligned}$$

In particular, the value of a variable-free term like $4+5\cdot 2$ does not depend on the state ν. In this case, the value is 14. The value of a term with variables, like $4 + x \cdot 2$, depends on ν. Suppose $\nu(x) = 5$, then $[\![4 + x \cdot 2]\!]_\nu = 14$. If $\omega(x) = 2$, then $[\![4 + x \cdot 2]\!]_\omega = 8$.

First-order Formulas. Unlike for terms, the value of a logical formula is not a number but instead *true* or *false*. Whether a logical formula evaluates to *true* or *false* depends on the interpretation of its symbols. In first-order logic of real arithmetic, the meaning of all symbols except variables is fixed. The meaning of terms and of formulas of first-order logic of real arithmetic is as usual in first-order logic, except that $+$ really means addition, $\cdot$ means multiplication, $\geq$ means greater or equals, and that the quantifiers $\forall x$ and $\exists x$ quantify over the reals.

Let ν_x^d denote the state that agrees with state ν except for the interpretation of variable x, which is changed to the value $d \in \mathbb{R}$:

$$\nu_x^d(y) = \begin{cases} d & \text{if } y \text{ is the variable } x\\ \nu(y) & \text{otherwise}\end{cases}$$

We write $\nu \models F$ to indicate that F evaluates to *true* in state ν and define it as follows.

Definition 7 (First-order logic semantics)**.** The *satisfaction relation* $\nu \models F$ for a first-order formula F of real arithmetic in state ν is defined inductively:

- $\nu \models (\theta_1 = \theta_2)$ iff $[\![\theta_1]\!]_\nu = [\![\theta_2]\!]_\nu$.
- $\nu \models (\theta_1 \geq \theta_2)$ iff $[\![\theta_1]\!]_\nu \geq [\![\theta_2]\!]_\nu$.
- $\nu \models \neg F$ iff $\nu \not\models F$, i.e. if it is not the case that $\nu \models F$.
- $\nu \models F \land G$ iff $\nu \models F$ and $\nu \models G$.
- $\nu \models F \lor G$ iff $\nu \models F$ or $\nu \models G$.
- $\nu \models F \to G$ iff $\nu \not\models F$ or $\nu \models G$.
- $\nu \models F \leftrightarrow G$ iff ($\nu \models F$ and $\nu \models G$) or ($\nu \not\models F$ and $\nu \not\models G$).
- $\nu \models \forall x\, F$ iff $\nu_x^d \models F$ for all $d \in \mathbb{R}$.
- $\nu \models \exists x\, F$ iff $\nu_x^d \models F$ for some $d \in \mathbb{R}$.

If $\nu \models F$, then we say that F is true at ν or that ν is a model of F. A formula F is *valid*, written $\vDash F$, iff $\nu \models F$ for all states ν. A formula F is a *consequence* of a set of formulas Γ, written $\Gamma \vDash F$, iff, for each ν: $\nu \models G$ for all $G \in \Gamma$ implies that $\nu \models F$.

With this definition, we know how to evaluate whether a evolution domain H of a continuous evolution $x' = \theta \,\&\, H$ is true in a particular state ν or not. If $\nu \models H$, then H holds in that state. Otherwise (i.e. if $\nu \not\models H$), H does not hold in ν. Yet, in which states ν do we need to check the evolution domain?

Continuous Programs. There is more than one way to define the meaning of a program, including defining a denotational semantics, an operational semantics, a structural operational semantics, an axiomatic semantics. For our purposes, what is most relevant is how a hybrid program changes the state of the system. Consequently, the semantics of HPs is based on which final states are reachable from which initial state. It considers which (final) state ω is reachable by running a HP α from an (initial) state ν. Semantical models that expose more detail, e.g., about the internal states during the run of an HP are possible [Pla10, Chapter 4] but not needed for our usual purposes.

If a differential equation starts in a state ν, the system could reach many possible states when following this particular differential equation. Even though the solutions of initial value problems (differential equation with an initial state) are unique under mild conditions (Appendix B), they still do not lead to a single unique state. Which state one ends up at when following a differential equation depends not only on the initial state ν, but also on how long the system follows that differential equation. Consequently, the meaning of a continuous program will invariably have to allow for many possible reachable states. Recall that $\mathcal{S}$ denotes the set of states.

The meaning of an HP α is given by a reachability relation $\rho(\alpha) \subseteq \mathcal{S} \times \mathcal{S}$ on states. That is, $(\nu, \omega) \in \rho(\alpha)$ means that final state ω is reachable from initial state ν by running HP α. From any initial state ν, there might be many states ω that are reachable, so many different ω for which $(\nu, \omega) \in \rho(\alpha)$. Form other initial states ν, there might be no reachable states ω at all for which $(\nu,\omega) \in \rho(\alpha)$. So $\rho(\alpha)$ is a proper relation, not a function.

Note 5. *The reachability relation $\rho(x' = \theta \,\&\, H)$ of a continuous program holds for all pairs of states that can be connected by a solution of the differential equation that is entirely within H:*

$$\begin{aligned}\rho(x' = \theta \,\&\, H) = \{(\varphi(0), \varphi(r)) \;:\; &\varphi(t) \models H \textit{ for all } 0 \le t \le r\\ &\textit{for a solution } \varphi : [0,r] \to \mathcal{S} \textit{ of } x' = \theta \textit{ of any duration } r \in \mathbb{R}\}.\end{aligned}$$

The first line in the definition of $\rho(x' = \theta \,\&\, H)$ means that the solution satisfies H at all times. The second line means that φ solves the differential equation, which essentially means that $\varphi(t) \models x' = \theta$ for all $0 \le t \le r$, when interpreting $\varphi(t)(x') \overset{\text{def}}{=} \frac{\mathsf{d}\varphi(\zeta)(x)}{\mathsf{d}\zeta}(t)$. Let us elaborate what this means and explicitly consider differential equation systems:

Definition 8 (Semantics of continuous programs)**.** $(\nu,\omega) \in \rho(x_1' = \theta_1, \dots, x_n' = \theta_n \,\&\, H)$ iff there is a *flow* φ of some duration $r \ge 0$ along $x_1' = \theta_1, \dots, x_n' = \theta_n \,\&\, H$ from state ν to state ω, i.e. a function $\varphi : [0,r] \to \mathcal{S}$ such that:

- $\varphi(0) = \nu, \varphi(r) = \omega$;
- φ respects the differential equations: For each variable x_i, the valuation $[\![x_i]\!]_{\varphi(\zeta)} = \varphi(\zeta)(x_i)$ of x_i at state $\varphi(\zeta)$ is continuous in ζ on $[0,r]$ and has a derivative of value $[\![\theta_i]\!]_{\varphi(\zeta)}$ at each time $\zeta \in (0,r)$;
- the value of other variables $z \notin \{x_1, \dots, x_n\}$ remains constant, that is, we have $[\![z]\!]_{\varphi(\zeta)} = [\![z]\!]_\nu$ for all $\zeta \in [0,r]$;
- and φ respects the invariant: $\varphi(\zeta) \models H$ for each $\zeta \in [0,r]$.

Observe that this definition is explicit about the fact that variables without differential equations do not change during a continuous program. The semantics of HP is *explicit change*: nothing changes unless (an assignment or) a differential equation specifies how. Also observe the explicit passing from syntax to semantics by the use of the valuation function $[\![\cdot]\!]$ in Def. 8.

A. Existence Theorems

For your reference, this appendix contains a short primer on some important results about differential equations [Pla10, Appendix B].

There are several classical theorems that guarantee existence and/or uniqueness of solutions of differential equations (not necessarily closed-form solutions with elementary functions, though). The existence theorem is due to Peano [Pea90]. A proof can be found in [Wal98, Theorem 10.IX].

Theorem 9 (Existence theorem of Peano)**.** *Let $f : D \to \mathbb{R}^n$ be a continuous function on an open, connected domain $D \subseteq \mathbb{R} \times \mathbb{R}^n$. Then, the initial value problem* (1) *with $(t_0, y_0) \in D$ has a solution. Further, every solution of* (1) *can be continued arbitrarily close to the boundary of D.*

Peano's theorem only proves that a solution exists, not for what duration it exists. Still, it shows that every solution can be *continued arbitrarily close to the boundary* of the domain D. That is, the closure of the graph of the solution, when restricted to $[0, 0] \times \mathbb{R}^n$, is not a compact subset of D. In particular, there is a global solution on the interval $[0, \infty)$ if $D = \mathbb{R}^{n+1}$ then.

Peano's theorem shows the existence of solutions of continuous differential equations on open, connected domains, but there can still be multiple solutions.

Example 10. The initial value problem with the following continuous differential equation

$$\begin{bmatrix} y' & = \sqrt[3]{|y|} \\ y(0) & = 0 \end{bmatrix}$$

has multiple solutions:

$$\begin{aligned} y(t) &= 0 \\ y(t) &= \left(\frac{2}{3}t\right)^{\frac{3}{2}} \\ y(t) &= \begin{cases} 0 & \text{for } t \leq s \\ \left(\frac{2}{3}(t-s)\right)^{\frac{3}{2}} & \text{for } t > s \end{cases} \end{aligned}$$

where $s \geq 0$ is any nonnegative real number.

B. Existence and Uniqueness Theorems

As usual, $C^k(D, \mathbb{R}^n)$ denotes the space of k times continuously differentiable functions from domain D to $\mathbb{R}^n$.

If we know that the differential equation (its right-hand side) is continuously differentiable on an open, connected domain, then the Picard-Lindelöf theorem gives a stronger result than Peano's theorem. It shows that there is a unique solution (except, of course, that the restriction of any solution to a sub-interval is again a solution). For this, recall that a function $f : D \to \mathbb{R}^n$ with $D \subseteq \mathbb{R} \times \mathbb{R}^n$ is called *Lipschitz continuous* with respect to y iff there is an $L \in \mathbb{R}$ such that for all $(t, y), (t, \bar{y}) \in D$,

$$\|f(t, y) - f(t, \bar{y})\| \leq L\|y - \bar{y}\|.$$

If, for instance, $\frac{\partial f(t,y)}{\partial y}$ exists and is bounded on D, then f is Lipschitz continuous with $L = \max_{(t,y)\in D} \|\frac{\partial f(t,y)}{\partial y}\|$ by mean value theorem. Similarly, f is *locally Lipschitz continuous* iff for each $(t, y) \in D$, there is a neighbourhood in which f is Lipschitz continuous. In particular, if f is continuously differentiable, i.e., $f \in C^1(D, \mathbb{R}^n)$, then f is locally Lipschitz continuous.

Most importantly, Picard-Lindelöf's theorem [Lin94], which is also known as the Cauchy-Lipschitz theorem, guarantees existence and uniqueness of solutions. As restrictions of solutions are always solutions, we understand uniqueness up to restrictions. A proof can be found in [Wal98, Theorem 10.VI]

Theorem 11 (Uniqueness theorem of Picard-Lindelöf)**.** *In addition to the assumptions of Theorem 9, let f be locally Lipschitz continuous with respect to y (for instance, $f \in C^1(D, \mathbb{R}^n)$ is sufficient). Then, there is a unique solution of the initial value problem (1).*

Picard-Lindelöf's theorem does not show the duration of the solution, but shows only that the solution is unique. Under the assumptions of Picard-Lindelöf's theorem, every solution can be extended to a solution of maximal duration arbitrarily close to the boundary of D by Peano's theorem, however. The solution is unique, except that all restrictions of the solution to a sub-interval are also solutions.

Example 12. The initial value problem

$$\begin{bmatrix} y' & = y^2 \\ y(0) & = 1 \end{bmatrix}$$

has the unique maximal solution $y(t) = \frac{1}{1-t}$ on the domain $t < 1$. This solution cannot be extended to include the singularity at $t = 1$.

The following global uniqueness theorem shows a stronger property when the domain is $[0, a] \times \mathbb{R}^n$. It is a corollary to Theorems 9 and 11, but used prominently in the proof of Theorem 11, and is of independent interest. A direct proof of the following global version of the Picard-Lindelöf theorem can be found in [Wal98, Proposition 10.VII].

Corollary 13 (Global uniqueness theorem of Picard-Lindelöf)**.** *Let $f : [0, a] \times \mathbb{R}^n \to \mathbb{R}^n$ be a continuous function that is Lipschitz continuous with respect to y. Then, there is a unique solution of the initial value problem (1) on $[0, a]$.*

Exercises

Exercise 1. Review the basic theory of ordinary differential equations and examples.

Exercise 2. Review the syntax and semantics of first-order logic.

References

[DGV96] Akash Deshpande, Aleks Göllü, and Pravin Varaiya. SHIFT: A formalism and a programming language for dynamic networks of hybrid automata. In Panos J. Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems*, volume 1273 of *LNCS*, pages 113–133. Springer, 1996.

[EEHJ96] Kenneth Eriksson, Donald Estep, Peter Hansbo, and Claes Johnson. *Computational Differential Equations*. Cambridge University Press, 1996.

[FKV04] G. K. Fourlas, K. J. Kyriakopoulos, and C. D. Vournas. Hybrid systems modeling for power systems. *Circuits and Systems Magazine, IEEE*, 4(3):16 – 23, quarter 2004.

[GBF+11] Radu Grosu, Grégory Batt, Flavio H. Fenton, James Glimm, Colas Le Guernic, Scott A. Smolka, and Ezio Bartocci. From cardiac cells to genetic regulatory networks. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *CAV*, volume 6806 of *LNCS*, pages 396–411. Springer, 2011. doi:10.1007/978-3-642-22110-1_31.

[Har64] Philip Hartman. *Ordinary Differential Equations*. John Wiley, 1964.

[KAS+11] BaekGyu Kim, Anaheed Ayoub, Oleg Sokolsky, Insup Lee, Paul L. Jones, Yi Zhang, and Raoul Praful Jetley. Safety-assured development of the gpca infusion pump software. In Samarjit Chakraborty, Ahmed Jerraya, Sanjoy K. Baruah, and Sebastian Fischmeister, editors, *EMSOFT*, pages 155–164. ACM, 2011. doi:10.1145/2038642.2038667.

[KGDB10] Branko Kerkez, Steven D. Glaser, John A. Dracup, and Roger C. Bales. A hybrid system model of seasonal snowpack water balance. In Karl Henrik Johansson and Wang Yi, editors, *HSCC*, pages 171–180. ACM, 2010. doi:10.1145/1755952.1755977.

[Lin94] M. Ernst Lindelöf. Sur l'application de la méthode des approximations successives aux équations différentielles ordinaires du premier ordre. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 114:454–457, 1894.

[LS10] Insup Lee and Oleg Sokolsky. Medical cyber physical systems. In Sachin S. Sapatnekar, editor, *DAC*, pages 743–748. ACM, 2010.

[LSC+12] Insup Lee, Oleg Sokolsky, Sanjian Chen, John Hatcliff, Eunkyoung Jee, BaekGyu Kim, Andrew L. King, Margaret Mullen-Fortino, Soojin Park, Alex Roederer, and Krishna K. Venkatasubramanian. Challenges and research directions in medical cyber-physical systems. *Proc. IEEE*, 100(1):75–90, 2012. doi:10.1109/JPROC.2011.2165270.

[PCA07] Leadership under challenge: Information technology R&D in a competitive world. an assessment of the federal networking and information technology R&D program. President's Council of Advisors on Science and Technology, Aug 2007. http://www.ostp.gov/pdf/nitrd_review.pdf.

[Pea90] Giuseppe Peano. Demonstration de l'intégrabilité des équations différentielles ordinaires. *Mathematische Annalen*, 37(2):182–228, 1890.

[PKV09] Erion Plaku, Lydia E. Kavraki, and Moshe Y. Vardi. Hybrid systems: from verification to falsification by combining motion planning and discrete search. *Form. Methods Syst. Des.*, 34(2):157–182, 2009.

[Pla07] André Platzer. Differential dynamic logic for verifying parametric hybrid systems. In Nicola Olivetti, editor, *TABLEAUX*, volume 4548 of *LNCS*, pages 216–232. Springer, 2007. doi:10.1007/978-3-540-73099-6_17.

[Pla08] André Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reas.*, 41(2):143–189, 2008. doi:10.1007/s10817-008-9103-8.

[Pla10] André Platzer. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Springer, Heidelberg, 2010. doi:10.1007/978-3-642-14509-4.

[Pla12] André Platzer. Logics of dynamical systems. In *LICS*, pages 13–24. IEEE, 2012. doi:10.1109/LICS.2012.13.

[Rei71] William T. Reid. *Ordinary Differential Equations*. John Wiley, 1971.

[RKR10] Derek Riley, Xenofon Koutsoukos, and Kasandra Riley. Reachability analysis of stochastic hybrid systems: A biodiesel production system. *European Journal on Control*, 16(6):609–623, 2010.

[Tiw11] Ashish Tiwari. Logic in software, dynamical and biological systems. In *LICS*, pages 9–10. IEEE Computer Society, 2011. doi:10.1109/LICS.2011.20.

[TPS98] Claire Tomlin, George J. Pappas, and Shankar Sastry. Conflict resolution for air traffic management: a study in multi-agent hybrid systems. *IEEE T. Automat. Contr.*, 43(4):509–521, 1998.

[Wal98] Wolfgang Walter. *Ordinary Differential Equations*. Springer, 1998.