The Poincaré-Bendixson Theorem in Isabelle/HOL

Fabian Immler*
Computer Science Department
Carnegie Mellon University
USA
fimmler@cs.cmu.edu

Yong Kiam Tan*
Computer Science Department
Carnegie Mellon University
USA
yongkiat@cs.cmu.edu

Abstract

The Poincaré-Bendixson theorem is a classical result in the study of (continuous) dynamical systems. Colloquially, it restricts the possible behaviors of planar dynamical systems: such systems cannot be chaotic. In practice, it is a useful tool for proving the existence of (limiting) periodic behavior in planar systems. The theorem is an interesting and challenging benchmark for formalized mathematics because proofs in the literature rely on geometric sketches and only hint at symmetric cases. It also requires a substantial background of mathematical theories, e.g., the Jordan curve theorem, real analysis, ordinary differential equations, and limiting (long-term) behavior of dynamical systems.

We present a proof of the theorem in Isabelle/HOL and highlight the main challenges, which include: *i)* combining large and independently developed mathematical libraries, namely the Jordan curve theorem and ordinary differential equations, *ii)* formalizing fundamental concepts for the study of dynamical systems, namely the α, ω-limit sets, and periodic orbits, *iii)* providing formally rigorous arguments for the geometric sketches paramount in the literature, and *iv)* managing the complexity of our formalization throughout the proof, e.g., appropriately handling symmetric cases.

CCS Concepts • **Mathematics of computing → Ordinary differential equations**; • **Theory of computation → Logic and verification**.

*This material is based upon work supported by the Air Force Office of Scientific Research under grant number FA9550-18-1-0120. Any opinions, finding, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force. The second author was also supported by A*STAR, Singapore. The authors are listed in alphabetical order.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CPP '20, January 20–21, 2020, New Orleans, LA, USA
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7097-4/20/01…$15.00
https://doi.org/10.1145/3372885.3373833

Keywords formalization of mathematics, dynamical systems, Poincaré-Bendixson theorem

ACM Reference Format:
Fabian Immler and Yong Kiam Tan. 2020. The Poincaré-Bendixson Theorem in Isabelle/HOL. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP '20), January 20–21, 2020, New Orleans, LA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3372885.3373833

1 Introduction

The qualitative study of ordinary differential equations was initiated by the seminal work of Poincaré [27]. The key idea is to study the behavior of ordinary differential equations by analyzing the differential equations themselves instead of solving them explicitly. This qualitative study is at the root of (continuous) dynamical systems theory [14], especially in the study of limiting (long-term) behavior of systems specified by differential equations.

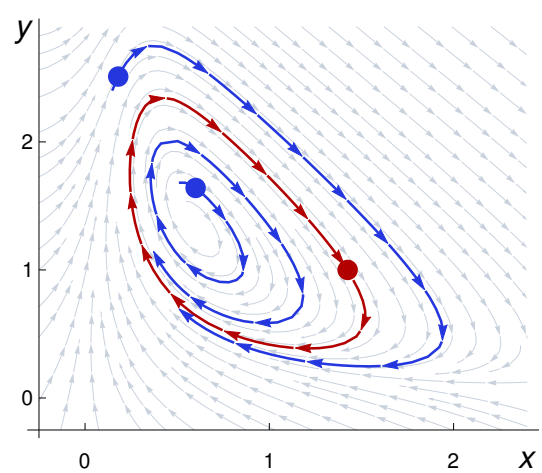


Figure 1. A visualization of Sel'kov's model of glycolysis $\dot{x} = -x + ay + x^2y, \dot{y} = b - ay - x^2y$ for the parameter values $a = 0.08$ and $b = 0.6$ [30, 32]. Forward trajectories from three selected points are shown in color.

Differential equations in the plane can be visualized by plotting their associated vector fields. Following Poincaré, the goal is then to deduce properties of the differential equations directly from geometric properties of the plot. For example, Fig. 1 visualizes Sel'kov's differential equations model for the biochemical process of glycolysis [30, 32]. Intuitively,

the arrows in Fig. 1 visualize the local direction in which solutions following the differential equations must travel. By locally (and continuously) flowing along these arrows, points trace out trajectories in the plane, such as the colored ones in Fig. 1. From the visualization, one might *hypothesize* that Sel'kov's model exhibits limiting *periodic* behavior, e.g., observe that the trajectory from the red point loops back onto itself (i.e., it is periodic), while the trajectories from the blue points tend towards the red trajectory asymptotically. This, in turn, provides a mathematical explanation for oscillations observed in the real world glycolysis process.

Yet, this simple visualization belies the difficulty of mathematically *proving* that the periodic behavior actually exists, and is not an artifact of inaccuracies in the visualization tool. The classical analytic tool that can be used to establish the existence of periodic behavior is the Poincaré-Bendixson theorem, named after Henri Poincaré [27] and Ivar Bendixson [4]. In a nutshell, the theorem asserts that the situation shown in Fig. 1 is the norm for planar dynamical systems: trajectories must either be periodic or tend to a trajectory that is periodic.[1] Notably, the theorem does *not* hold in higher dimensions where more complicated behavior is possible.

We formalize the Poincaré-Bendixson theorem in the Isabelle/HOL proof assistant [23, 24], drawing on material presented in several textbooks [6, 7, 9, 25, 28, 33, 34]. Our proof of the theorem itself mainly follows Coddington and Levinson [7], Dumortier, Llibre and Artés [9], and Perko [25]. Beyond its applications in formalizing dynamical systems theory, the theorem is an interesting and challenging benchmark for formalized mathematics, because:

1. It requires a mature analysis library, e.g., the proof makes central use of the Jordan curve theorem[2] and the theory of ordinary differential equations.
2. Proofs in the mathematical literature rely heavily on geometric intuition and arguing for symmetric cases without loss of generality.

Our formalization meets the first challenge:

i) It builds on existing work in Isabelle/HOL, namely the Jordan curve theorem and the theory of ordinary differential equations (Section 2). Isabelle/HOL's (real) analysis libraries are also used extensively.

ii) It provides a new library of fundamental dynamical systems concepts (Section 3). This library contains definitions of α, ω-limit sets and periodic orbits, and proofs of their standard mathematical properties.

Our formalization also meets the second challenge:

iii) We prove the Poincaré-Bendixson theorem (Section 4) as stated in Coddington and Levinson [7, Thms. 2.1, 3.1]. Our proof formalizes the first (as far as we know) fully rigorous argument for an important geometric lemma (Section 4.2) — this lemma makes fundamental use of the Jordan curve theorem and is usually argued based on geometric sketches in textbooks [7, 9, 25]. Our argument is inspired by the gate theorem [3].

iv) We report on a number of formalization techniques used throughout the proof (Section 5), notably our use of locales [2] to avoid duplication while reasoning about symmetric cases for the forward and backward time trajectories of dynamical systems.

As an application, we use the theorem to prove the existence of limiting periodic behavior for two examples (Section 6), including the instance of Sel'kov's model in Fig. 1.

The formalization is ≈7000 lines. It is available in the Archive of Formal Proofs [20] and works with Isabelle2019.[3] All definitions and theorems formalized in Isabelle/HOL are typeset in `typewriter` font and with boldface **keywords**. Explanations of formalized arguments also use `typewriter` font. Regular typesetting is reserved for informal arguments.

2 Background

Our formalization builds on the existing libraries for analysis and ordinary differential equations [10, 13, 15, 18, 19, 21] in Isabelle/HOL and the Archive of Formal Proofs. This section recalls relevant concepts from these libraries.

2.1 Analysis

This section briefly reviews the most important notation that is used throughout the paper. Infix ‘ is Isabelle/HOL's notation for the image of a function applied to a set, i.e.:

```
f ‘ X = { f x | x ∈ X }
```

In Isabelle/HOL's analysis library, limits are formalized generically using filters [15] (see Section 5 for more detail). Two kinds of limits are used frequently in the formalization. First, a convergent sequence `s` tending to limit `l` is written as `s ⟶ l`. For our formalization's purposes, it can be unfolded to its usual (real) analytic definition as follows:

```
lemma tendsto_sequentially:
 "(s ⟶ l) ⟷
   (∀e>0. ∃N. ∀n≥N. dist (s n) l < e)"
```

Second, for (divergent) real-valued sequences `s`, the sequence diverging to positive (resp. negative) infinity is written as `s ⟶ ∞` (resp. `s ⟶ −∞`). These similarly obey the standard unfoldings from mathematical analysis:

```
lemma filterlim_at_top_sequentially:
 "(s ⟶ ∞) ⟷ (∀w. ∃N. ∀n≥N. s n ≥ w)"
and filterlim_at_bot_sequentially:
 "(s ⟶ -∞) ⟷ (∀a. ∃N. ∀n≥N. s n ≤ a)"
```

Both limits are actually defined using Isabelle/HOL's notion of generalized limit `filterlim` as shown below. The formalization mostly uses theorems for the general `filterlim` constant when reasoning about these limits.

[1] The precise conditions under which this is true is given later in the paper.
[2] Technically, the proof only needs the theorem for piecewise smooth curves, but the theorem is available in full generality in Isabelle/HOL.
[3] https://isabelle.in.tum.de/

```
lemma filterlim_equalities:
 "(s ⟶ l) ⟷ filterlim s (nhds l) sequentially"
 "(s ⟶ ∞) ⟷ filterlim s at_top sequentially"
 "(s ⟶ -∞) ⟷ filterlim s at_bot sequentially"
```

2.2 Ordinary Differential Equations

The theory of ordinary differential equations (ODEs) in Isabelle/HOL was formalized by Immler *et al.* [18, 19, 21]. Their formalization includes, for the study of dynamical systems, the all-important notion of the *flow* of a dynamical system [21]. Key properties of the flow are recalled below.

Everything in this paper is developed under the assumption that the given (autonomous) ODE has right-hand side (RHS) $f : X \to X$ where X is an open subset of Euclidean space $\mathbb{R}^n$ and f is a continuously differentiable function.

$$\dot{x} = f(x) \tag{1}$$

These assumptions guarantee that the differential equations (1) have a *unique* solution $\phi : X \times \mathbb{R} \to X$, which is henceforth called the *flow* of the ODE [6]. The value of the solution to the differential equation at time t for a given initial condition $x(0) = x_0 \in X$ is given by the flow: $\phi(x_0, t)$. A technical intricacy in the formalization (often elided in this paper) is that the flow from a particular initial condition $\phi(x_0, \cdot)$ may not be defined for all times $t \in \mathbb{R}$, but only on some non-empty, *maximal*, open existence interval [6].

In Isabelle/HOL's library for ODEs, the global assumption of a continuously differentiable RHS f is formalized as a locale [2] with parameters `f`,`f'` for the ODE and `X` for the domain of `f` as follows:

```
lemma c1_on_open_def: "c1_on_open f f' X ⟷
  open X ∧
  (∀x. x ∈ X ⟶ (f has_derivative f'(x)) (at x)) ∧
  continuous_on X f'"
```

Here, `f'(x)` is a linear function and continuity of `f'` is with respect to the operator norm. The locale `c1_on_open f f' X` is used as the context throughout our formalization. In this context, the ODE library provides the notion of flow (`flow0`) and its associated existence interval (`existence_ivl0`) as the main interfaces. The characteristic property of `flow0 x0` is that it obeys the ODE at all times `t` within its associated existence interval (`existence_ivl0 x0`):

```
lemma flow_has_vector_derivative:
 "t ∈ existence_ivl0 x0 ⟹
  (flow0 x0 has_vector_derivative
    (f (flow0 x0 t))) (at t)"
```

Additional properties that are used (but not presented here) are uniqueness and continuity of the flow and the fact that `existence_ivl0 x0` is indeed the maximal interval of existence for the solution from initial point `x0`.

A fundamental property of dynamical systems is that the description of its evolution (i.e., the flow) is a group action of the real numbers (time `t`) on the state `x`. More concretely, this means that when `0` time passes, the system does not evolve, and evolving for time `s + t` must be the same as evolving first for time `s` and then for time `t`:

```
lemma flow_group_action:
 "x0 ∈ X ⟹ flow0 x0 0 = x0"
and
 "s ∈ existence_ivl0 x0 ⟹
  t ∈ existence_ivl0 (flow0 x0 s) ⟹
  flow0 x0 (s + t) = flow0 (flow0 x0 s) t"
```

2.2.1 Time Reversal

The *time reversal* property of the flow is often exploited to symmetrically reason about forward and backward time but restricting the attention *without loss of generality* to only the forward time case. For the moment, we make the dependency of flow on differential equation `f` explicit and write $\texttt{flow0}_{\texttt{f}}$ for the flow of the ODE with RHS `f`. It is a theorem that when negating the ODE's RHS (think reversing the arrows in Fig. 1), the flow of the reversed ODE $\texttt{flow0}_{\texttt{(-f)}}$ corresponds to the flow of the original ODE in backward time `-t`:

```
lemma flow_eq_rev: "flow0(-f) x0 t = flow0f x0 (-t)"
```

Reversing the ODE also flips the existence interval:

```
lemma existence_ivl_eq_rev:
 "t ∈ existence_ivl0(-f) x ⟷
   -t ∈ existence_ivl0f x0"
```

These equations underpin the time reversal reasoning used throughout Sections 3 and 4. The technical detail for how this is achieved with minimal proof effort is in Section 5.3.

2.3 Jordan Curve Theorem

The Jordan curve theorem is formalized in Isabelle/HOL for the complex plane since it is mostly used in the complex analysis libraries. The real plane is more natural for the setting of the Poincaré-Bendixson theorem. In our formalization, the real plane is represented, as generally as possible, with a type `'a` under the type class constraint `'a::euclidean_space` and a dimension assumption `DIM('a) = 2`. This way, depending on the application, `'a` can be instantiated, e.g., with pairs of real numbers (`real*real`), 2-vectors of real numbers (`real^2`), or complex numbers (`complex`). The following version of the Jordan curve theorem for the real plane is proved straightforwardly from the existing Jordan curve theorem using the obvious bijection to the complex numbers:

```
lemma Jordan_curve_R2:
fixes c :: "real ⇒ 'a"
assumes "simple_path c" "pathfinish c = pathstart c"
obtains inside outside where
 "inside ≠ {}" "open inside" "connected inside"
 "outside ≠ {}" "open outside" "connected outside"
 "bounded inside" "¬ bounded outside"
 "inside ∩ outside = {}"
 "inside ∪ outside = - path_image c"
 "frontier inside = path_image c"
 "frontier outside = path_image c"
```

Using this theorem is heavyweight: one first needs to construct a simple path `c`, i.e., a curve in the plane with no self-crossings. Then, the conclusion yields two sets `inside` and `outside`, with the former being the points inside the curve and the latter being the points outside, see Fig. 5 for a visualization. More importantly, `inside` and `outside` come with 12 characterizing properties[4] that must be tracked and used together when working with those sets in proofs.

3 Limit Sets and Periodic Orbits

This section explains our definitions of standard concepts in the study of dynamical systems, as well as proofs of their key properties. Our formalization mostly follows the mathematical literature except opting for the most general (or least restrictive) definitions where possible. The contents of this section are not restricted to the plane.

3.1 Limit Sets

Intuitively, the ω-limit set of a point x, is the set of points that the flow from x tends to in positive time. The ω-limit set is *not* simply the limit $\lim_{t\to\infty} \phi(x,t)$. As Fig. 1 shows, this limit does not even exist for any of the colored trajectories. Instead, the colored trajectories tend to an entire set of points, namely, those on the red periodic orbit. We define ω-limit points (and sets) as follows:

```
lemma ω_limit_point_def: "ω_limit_point x p ⟷
  {0..} ⊆ existence_ivl0 x ∧
  (∃s. s ⟶ ∞ ∧ (flow0 x ∘ s) ⟶ p)"
lemma ω_limit_set_def:
 "ω_limit_set x = {p. ω_limit_point x p}"
```

Here, `p` is a ω-limit point of `x` if there is a sequence of times `s` tending to infinity where the flow evaluated at those times tends to `p`. Additionally, `{0..} ⊆ existence_ivl0 x` ensures that the existence interval for `x` extends to ∞ so that the sequence `flow0 x ∘ s` is well-defined. The definition of ω-limit sets includes ω-limit points `p` that are on the boundary of the domain `X` of the ODEs. Including these points makes the definition as general as possible: one can explicitly exclude these boundary points later, if desired. The choice of whether to include these points depends on the application, e.g., [25, Chap. 3.2, Def. 1] excludes them, [33, §30.V] includes them, and for [6, Def. 1.165] there is no difference because the domain `X` is assumed to be $\mathbb{R}^n$, which is unbounded.

Several properties of the ω-limit set (further below) use the following notion of the flow from a point `x` being trapped forward in time on set `K` on its positive existence interval:

```
lemma trapped_forward_def: "trapped_forward x K ⟷
  (flow0 x ` (existence_ivl0 x ∩ {0..}) ⊆ K)"
```

If all points in `K` are trapped forward in `K` itself, then `K` is called a *positively invariant* set:

```
lemma positively_invariant_def:
  "positively_invariant K ⟷
    (∀x∈K. trapped_forward x K)"
```

The definitions (not shown here) of `trapped_backward` and `negatively_invariant` are similar, but with respect to negative existence intervals (`existence_ivl0 x ∩ {..0}`) instead.

The ω-limit set is closed and invariant (i.e., both positively and negatively invariant) [6, Prop. 1.167]:

```
lemma ω_limit_set_closed: "closed (ω_limit_set x)"
lemma ω_limit_set_invariant:
  "invariant (ω_limit_set x)"
```

The ω-limit set for a point `x` whose flow is trapped forward in a compact set `K` enjoy additional properties: it is *i)* non-empty, *ii)* a subset of `K` (hence itself compact), *iii)* connected, and *iv)* the flow from points in the ω-limit set exists globally [6, Prop. 1.168]. These four properties are formalized below, where the group of assumptions `xK` says that the flow from `x` is trapped forward in the compact set `K`:

```
lemma ω_limit_set_in_compact:
assumes xK: "compact K" "K ⊆ X" "x ∈ X"
 "trapped_forward x K"
shows
 "ω_limit_set x ≠ {}"
 "ω_limit_set x ⊆ K"
 "connected (ω_limit_set x)"
 "⋀y. y ∈ ω_limit_set x ⟹ existence_ivl0 y = UNIV"
```

The α-limit set of `x` is the set of points that the flow from `x` tends to in negative time. Thanks to the symmetry in time, α-limit sets of a flow are simply ω-limit sets of the time-reversed flow. Thus, instead of reproving analogous results for the α-limit set, we reuse the results for the ω-limit set of the reverse flow and rewrite with the equations that relate limit sets of the reverse flow. Concretely, making the dependency on the ODE `f` explicit as in Section 2.2.1, these are the relevant rewriting equations:

```
lemma ω_α_limit_eq_rev:
 "ω_limit_point(-f) x p = α_limit_pointf x p"
 "α_limit_point(-f) x p = ω_limit_pointf x p"
 "ω_limit_set(-f) x = α_limit_setf x"
 "α_limit_set(-f) x = ω_limit_setf x"
```

3.2 Closed and Periodic Orbits

Periodic orbits are of special interest in dynamical systems because they provide canonical examples of oscillatory behavior in systems. We define the slightly more general *closed* orbits as those that return to the initial point in non-zero time; periodic orbits are closed orbits where the minimal such time (i.e., the *period*) is non-zero:[5]

```
lemma closed_orbit_def: "closed_orbit x ⟷
  (∃T ∈ existence_ivl0 x. T ≠ 0 ∧ flow0 x T = x)"
```

[4] Technically, the Poincaré-Bendixson theorem does not use the fact that `inside` is bounded while `outside` is unbounded.

[5] Technically, the word *orbit* usually refers to the entire set of points obtained by flowing a point `x` forward and backward in time. Our definition drops this distinction, allowing any point along this flow to be taken as a representative of the orbit.

```
lemma period_def: "period x =
  Inf {T ∈ existence_ivl0 x. T > 0 ∧ flow0 x T = x}"
lemma periodic_orbit_def: "periodic_orbit x ⟷
  closed_orbit x ∧ period x > 0"
```

The degenerate case with period zero corresponds to *equilibrium points* of the differential equations, i.e., points `x` where the RHS evaluates to zero (`f x = 0`). Points `x` where the RHS is non-zero `f x ≠ 0` are called *regular points*. The slightly more general definition of closed orbits allows us to prove properties of both equilibria and periodic orbits simultaneously. For example, for closed orbits, the flow exists globally for all times (`existence_ivl0 x = UNIV`) and the set of all points reached through the flow (`flow0 x ‘ UNIV`) is equal to both α- and ω-limit sets:

```
lemma closed_orbit_limit_sets:
assumes "closed_orbit x"
shows
 "existence_ivl0 x = UNIV"
 "flow0 x ‘ UNIV = ω_limit_set x"
 "flow0 x ‘ UNIV = α_limit_set x"
```

Nevertheless, equilibria can be distinguished from periodic orbits very simply. The point `x` is on a periodic orbit if and only if it is regular and on a closed orbit:

```
lemma closed_orbit_periodic:
 "periodic_orbit x ⟷ closed_orbit x ∧ f x ≠ 0"
```

4 The Poincaré-Bendixson Theorem

For the Poincaré-Bendixson theorem, the dynamics are assumed to be on the plane, i.e., `'a::euclidean_space` is a vector space with dimension `DIM('a) = 2`. We formalize two versions of the theorem, in increasing generality. The first (less general) version follows [7, Chap. 16, Thm. 2.1]:

```
theorem poincare_bendixson:
assumes xK: "compact K" "K ⊆ X" "x ∈ X"
 "trapped_forward x K"
assumes "0 ∉ f ‘ (ω_limit_set x)"
obtains y where
 "periodic_orbit y"
 "flow0 y ‘ UNIV = ω_limit_set x"
```

This theorem says that any point `x` whose flow is trapped in a compact set `K`, and whose ω-limit set contains no equilibrium points must tend towards a periodic orbit. In particular, this periodic orbit is *equal* to the ω-limit set of `x`. This form of the theorem is most commonly used in applications to establish the existence of periodic orbits (Section 6). Indeed, all three trajectories illustrated in Fig. 1 display the behavior asserted by this theorem. The ω-limit sets of all three colored trajectories have no equilibrium points and so by the theorem they must all tend to periodic orbits. The blue trajectories tend towards the red periodic orbit, and the red trajectory is itself a periodic orbit (so it tends to itself).

The second (more general) version follows [7, Chap. 16, Thm. 3.1]. It gives a description of the possibilities in case there are finitely many equilibrium points in `K`:

```
theorem poincare_bendixson_general:
assumes xK: "compact K" "K ⊆ X" "x ∈ X"
 "trapped_forward x K"
assumes "S = {y ∈ K. f y = 0}" "finite S"
shows
 "(∃y ∈ S. ω_limit_set x = {y}) ∨
  (∃y. periodic_orbit y ∧
    flow0 y ‘ UNIV = ω_limit_set x) ∨
  (∃P R. ω_limit_set x = P ∪ R ∧
    P ⊆ S ∧ 0 ∉ f ‘ R ∧ R ≠ {} ∧
    (∀z ∈ R.
      (∃p1 ∈ P. ω_limit_set z = {p1}) ∧
      (∃p2 ∈ P. α_limit_set z = {p2})))"
```

The three possibilities for the ω-limit set of `x`, corresponding to the three disjuncts respectively, are: *i)* it consists of a single equilibrium point, or *ii)* it is a periodic orbit, or *iii)* it consists of (finitely many) equilibrium points `P` and a non-empty set of regular points `R` such that every regular point in `R` tends to exactly one equilibrium point `p1 ∈ P` in forward time and one `p2 ∈ P` in backward time; note that `p1` and `p2` may be the same equilibrium point. The reader is referred to [9, Figs. 1.17-1.19] for an illustration of these three possibilities. The rest of this section focuses on the earlier `poincare_bendixson` theorem since it is used as a crucial stepping stone in the proof of `poincare_bendixson_general` [7], and most of the important lemmas are shared between both theorems. We give an outline of the proof and then focus on one particularly challenging step in the formalization.

4.1 Proof Outline

Our proof outline largely follows Perko [25]. The proof makes heavy use of *transversal segments*. These are two-dimensional line segments specified by two end points `a,b` such that the RHS `f` of the differential equation along the segment is nowhere parallel to it. Below, `{a--b}` is notation for the closed line segment between points `a,b`, `{a<--<b}` denotes the open line segment (excluding the endpoints `a,b`), `·` is the inner product, and `rot` denotes the 90 degree rotation of a (2D) vector. A transversal segment is illustrated in Fig. 2.

```
lemma transversal_segment_def:
 "transversal_segment a b ⟷
    a ≠ b ∧ {a--b} ⊆ X ∧
    (∀z ∈ {a--b}. f z · rot (a-b) ≠ 0)"
```

Proof Sketch. We begin by enumerating several key lemmas used in the proof. The main proof is at ⑤.

① A transversal segment always exists around any regular point by continuity of `f`, see Fig. 2 for an illustration:

```
lemma transversal_segment_exists:
assumes "x ∈ X" "f x ≠ 0"
obtains a b where "x ∈ {a<--<b}"
  "transversal_segment a b"
```

② If `p` is an ω-limit point on a transversal segment, then, in addition to having a sequence of times `s` tending to ∞ where the flow tends to `p` (which is the definition of ω-limit points),

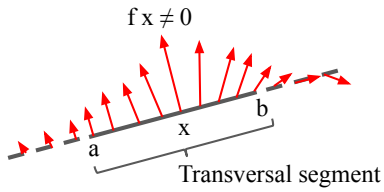


Figure 2. A transversal segment between points a, b containing the regular point x with f x ≠ 0. The RHS of the ODEs f (represented by red arrows) is never parallel to the segment (or zero) at points on the segment between a, b (inclusive).

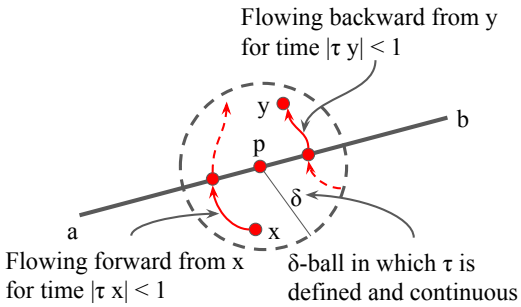


Figure 3. The map τ obtained from the implicit function theorem flows points x, y near p onto the transversal segment.

s can be chosen so that the flow at those points also lie on the transversal segment:

```
lemma ω_limit_crossings:
assumes "transversal_segment a b"
assumes "ω_limit_point x p"
assumes "p ∈ {a<--<b}"
obtains s where
 "s ⟶ ∞" "(flow0 x ∘ s) ⟶ p"
 "∀n. flow0 x (s n) ∈ {a<--<b}"
```

We now elaborate on the formal proof of this lemma. By definition, since p is an ω-limit point there is a sequence of times t at which the flow converges to p:

```
obtain t where
 "t ⟶ ∞" "(flow0 x ∘ t) ⟶ p"
```

The idea is to modify the times t slightly, so that the sequence still converges to p, but does so *on* the transversal segment instead. The mathematical tool that enables this slight modification is the implicit function theorem.[6] When applied to the flow of the ODE, it yields a continuous function τ whose value at a point y is the time it takes for the flow from y to return to the transversal segment. The function is well-defined and continuous for points in a δ-ball around p. The return time can be negative, but is bounded in norm, see Fig. 3 for an illustration. The formal property about p obtained from the implicit function theorem is as follows:

[6]This is similar to the formalization of the first return time of the Poincaré map by Immler and Traut [21].

```
obtain τ δ
where "0 < δ" "continuous_on (ball p δ) τ"
 "τ p = 0" "(⋀y. y ∈ ball p δ ⟹ |τ y| < 1)"
 "(⋀y. y ∈ ball p δ ⟹ flow0 y (τ y) ∈ {a<--<b})"
```

The modified sequence of times s is defined by adding the return time τ for the flow at each time in t:

```
define s where "s n = t n + τ (flow0 x (t n))" for n
```

Note that s diverges because τ is bounded in norm. Rewriting with the semi-group property of the flow yields:

```
have "∀n. (flow0 x o s) n =
  flow0 ((flow0 x o t) n) ((τ o (flow0 x o t)) n)"
```

From this and the facts that the flow at times t converges to p, and τ is continuous at p, it follows that the flow at times along the modified sequence converges to flow0 p (τ p):

```
from ‹(flow0 x ∘ t) ⟶ p› and ‹τ −p→ τ p›
have
 "(λn. flow0 ((flow0 x ∘ t) n) ((τ ∘ (flow0 x ∘ t)) n))
  ⟶
  flow0 p (τ p)"
```

Moreover, the specification of the return time map τ guarantees that flow0 p (τ p) = p and that the flow at every time in s is on the segment, which completes the proof of ②.

③ Transversal segments contain at most one ω-limit point. The proof of this crucial lemma is deferred since it depends fundamentally on the dynamical system being planar and its proof requires the Jordan curve theorem. It also leads to an interesting formalization challenge (Section 4.2).

```
lemma unique_transversal_segment_intersection:
assumes "transversal_segment a b"
assumes "u ∈ ω_limit_set x ∩ {a<--<b}"
shows "ω_limit_set x ∩ {a<--<b} = {u}"
```

④ If the ω-limit set contains a periodic orbit, then it is equal to the periodic orbit. The proof of this step is technical and can be found in the cited textbooks. It uses ③ and connectedness of the ω-limit set.

```
lemma periodic_imp_ω_limit_set:
assumes xK: "compact K" "K ⊆ X" "x ∈ X"
 "trapped_forward x K"
assumes "periodic_orbit y"
assumes "flow0 y ‘ UNIV ⊆ ω_limit_set x"
shows "flow0 y ‘ UNIV = ω_limit_set x"
```

⑤ The proof of poincare_bendixson is straightforward using lemmas ①–④. Let y be a point in the ω-limit set of x and consider a point z in the ω-limit set of y. Note that z is also in the ω-limit set of x by invariance (and closure) of the ω-limit set. By assumption, both y, z are regular points. By ①, there is a transversal segment through z. By ②, since z is on the transversal segment and in the ω-limit set of y, there is a sequence of times s such that the flow from y at those times tends to z and is also on the transversal segment. Pick any two distinct times t1, t2 in s; the flow at these times (call them x1, x2) are both in the ω-limit set of x. By ③, x1, x2 must be identical points, which implies that the flow from y

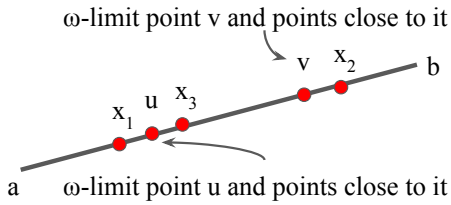


Figure 4. The (impossible) situation with two distinct ω-limit points u, v on the transversal segment.

is a periodic orbit. By ④, since y is a periodic orbit contained in the ω-limit set of x, it is equal to the ω-limit set of x. □

Returning to the proof of ③, suppose for contradiction that there are are two distinct ω-limit points u, v on the transversal segment. By ②, pick three times along the flow $t_1 < t_2 < t_3$ such that the flow lies on the transversal segment at those times. Respectively denote the flow at these time as x_1, x_2, x_3. The times can be chosen such that x_1, x_3 are close to u, while x_2 is close to v. A configuration of these points along the transversal segment is illustrated in Fig. 4.

However, for planar systems, the points x_1, x_2, x_3 must be arranged monotonically in the order of the segment because of the upcoming monotonicity lemma in Section 4.2. Hence, the situation described above is impossible, contradiction.

4.2 The Monotonicity Lemma

This monotonicity property is fundamental to the plane and its proof requires the Jordan curve theorem. We present both a textbook proof sketch and our formalized argument in order to highlight the important differences.

4.2.1 A Textbook Proof

The following is a brief textbook statement and proof sketch of the monotonicity lemma.

Lemma 4.1 (Monotonicity of intersections [7, 9, 25])**.** *If the flow on a closed time interval intersects a transversal segment, then it does so in a finite number of points. These intersections are monotonic in the order of the transversal segment.*

Proof Sketch. First, by standard analytic arguments, there can only be a finite number n of times where the image of the flow from a point on a closed time interval intersects the transversal segment [7, 9, 25] (see Section 5.5). Order these times $t_1 < t_2 < \cdots < t_n$, and refer to their respective intersection points with the transversal segment as $x_1, x_2, \ldots, x_n$. Consider the first two successive intersections x_1, x_2 at times t_1, t_2. Without loss of generality, assume that $x_1 < x_2$ according to the order of the transversal segment. This situation is sketched in Fig. 5, where the blue curve represents the flow and the black line between a, b is a transversal segment. Along this transversal segment, the flow must cross at x_1, x_2 (and also at subsequent intersections, e.g., x_3) in the same direction; this is illustrated by the red directional arrows.

Since the flow does not intersect with the transversal segment between times t_1, t_2, the (dashed green) curve formed by the flow between these times and the line segment between x_1, x_2 forms a Jordan curve J. From the Jordan curve theorem, J divides the plane into an inside (I, shaded and striped green) and outside (O, unshaded). From Fig. 5, two cases are possible, depending on the flow. In case Ⓐ, the flow at times $t > t_2$ locally lies in I while in case Ⓑ it lies locally in O. In fact, in case Ⓐ (resp. Ⓑ), the flow must remain in I (resp. O) for *all* times $t > t_2$ because it cannot touch or cross the Jordan curve J by construction. Therefore, any further intersections at time $t_3 > t_2$ must occur at a point x_3 which is beyond x_2 according to the order of the segment.

This argument extends by induction to the ordered list of times $t_1 < t_2 < \cdots < t_n$, and therefore the intersections $x_1, x_2, \ldots x_n$ are ordered monotonically in the same order as $x_1 < x_2$ along the transversal segment. □

Formalizing this proof sketch is challenging for two primary reasons. First, textbook proofs (rightfully) elide several symmetric cases, e.g., assuming without loss of generality that $x_1 < x_2$ (and $t_1 < t_2$), and handling only one of the Ⓐ and Ⓑ cases arising from the Jordan curve theorem.[7] Second, the sketches in Fig. 5 may be convincing to a human reader, but less so for a skeptical proof assistant. Our proof proceeds similarly in two steps, first proving the case of two successive intersections, before extending to the general case.

4.2.2 A Formal Proof (Successive Intersections)

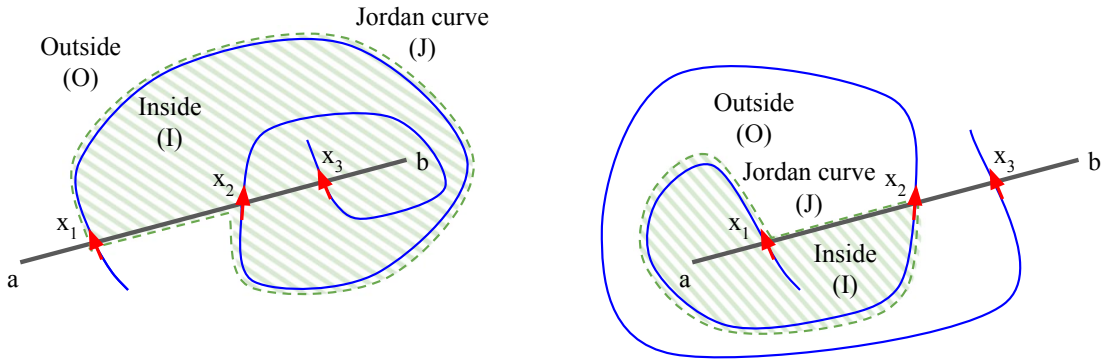
Applying the Jordan curve theorem is heavyweight and it is desirable to isolate its usage to a single lemma, obtaining other required cases by symmetry. Our main lemma for successive intersections formalizes the particular case described in the proof sketch above. Formally:

```
lemma flow0_transversal_segment_monotone_step:
assumes "transversal_segment a b"
assumes "t1 ≤ t2" "{t1..t2} ⊆ existence_ivl0 x"
assumes x1: "flow0 x t1 ∈ {a<--<b}"
assumes x2: "flow0 x t2 ∈ {flow0 x t1<--<b}"
assumes "⋀t. t ∈ {t1<..<t2} ⟹ flow0 x t ∉ {a<--<b}"
assumes "t > t2" "t ∈ existence_ivl0 x"
shows "flow0 x t ∉ {a<--<flow0 x t2}"
```

The assumptions `x1`, `x2` say the flow from `x` at times `t1`, `t2` are on the transversal segment, and that `flow0 x t2` is after `flow0 x t1` according to the order of the transversal segment. For brevity, we refer to `flow0 x t1` as `x1` and `flow0 x t2` as `x2` from now on. The assumption `⋀t. t ∈ {t1<..<t2} ⟹ . . .` says that no other transversal segment intersections occur between times `t1`, `t2`. Following convention, the subscripts t_1, t_2, x_1, x_2 are typeset as `t1`, `t2`, `x1`, `x2` throughout.

The primary departure in our formalization of this lemma from standard textbook proofs lies after the construction of

[7] In fact, several textbook arguments omit (without even mentioning the possibility of) one of the cases.



Case Ⓐ: flow remains inside J for times $t > t_2$.

Case Ⓑ: flow remains outside J for times $t > t_2$.

Figure 5. A sketch of the two cases that arise from applying the Jordan curve theorem. Note that the Jordan curve (in dashed green) is slightly offset for clarity in the illustration. It should lie exactly on the blue curve and line segment between x_1, x_2.

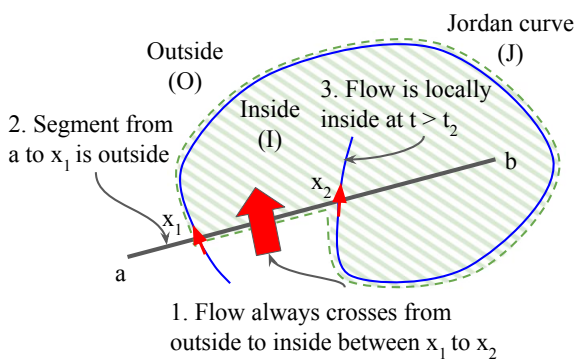


Figure 6. The three pieces of information that must be established simultaneously for case Ⓐ.

the Jordan curve `J`. Here, we must show that the two cases Ⓐ and Ⓑ depicted in Fig. 5 are, in fact, the only ones that can occur. The difficulty is that the proof must simultaneously establish three pieces of information in the two cases[8]:

1. At points along the segment from `x1` to `x2`, for Ⓐ, the (local) flow from those points crosses from the outside `O` to the inside `I`. For Ⓑ, they cross from `I` to `O`.
2. The segment from `a` to `x1` is, for Ⓐ, entirely in `O`. For Ⓑ, it lies entirely in `I`.
3. For all times `t > t2` the flow from `x`, for Ⓐ, is (locally) in `I`. For Ⓑ, the flow is in `O`.

An illustration of the three pieces of information for case Ⓐ is shown in Fig. 6. We now sketch our construction that establishes these three pieces of information rigorously and that is amenable to formalization. This construction is of wider interest since it is, to the best of our knowledge, new.

[8]Equivalently, it must establish that no other combination of facts are possible, e.g., having the flow after `t > t2` locally in `I` while simultaneously having the segment from `a` to `x1` lie in `I`.

Proof Sketch. The proof simultaneously establishes all three pieces of information by what we call a *flow region* construction. The key analytic step behind this construction is to find a lower bound `T > 0` for which the flow from points on the transversal segment at time `0` cannot return to the segment in positive `0 < s ≤ T` (or in negative `-T ≤ s < 0`) time. Geometrically, `rot n` is a normal vector pointing in the same direction as the flow along the transversal segment, so that `(y - x) · rot n > 0` is true for points `y` that are "above" the transversal segment and `(y - x) · rot n < 0` is true for points that are "below" it:

```
lemma leaves_transversal_segmentE:
assumes transversal: "transversal_segment a b"
obtains T n where "T > 0" "n = a - b ∨ n = b - a"
 "⋀x. x ∈ {a -- b} ⟹ {-T..T} ⊆ existence_ivl0 x"
 "⋀x s. x ∈ {a -- b} ⟹ 0 < s ⟹ s ≤ T ⟹
    (flow0 x s - x) · rot n > 0"
 "⋀x s. x ∈ {a -- b} ⟹ -T ≤ s ⟹ s < 0 ⟹
    (flow0 x s - x) · rot n < 0"
```

This analytic lemma is proved using the compactness of the transversal segment. The normal vector `rot n` is used to factor handling of symmetric cases into the handling of the disjunction `n = a - b ∨ n = b - a`.

The first region `r1` is constructed by locally flowing points on the open segment between `x1` to `b` forward for an open time interval `(0,s)` with `0 < s ≤ T`. The second region `r2` is constructed by similarly locally flowing the open segment between `a` to `x2` backward. The resulting (open) flow regions are illustrated for case Ⓐ in Fig. 7. Using the time bound `T`, both regions `r1,r2` are chosen so that they do not intersect with the Jordan curve `J`. Moreover, these regions are (path) connected by construction. Thus, if there is a point in `r1` that is contained in `I` (resp. in `O`), then all of `r1` is in contained `I` (resp. in `O`). The same is true for region `r2`.

The regions `r1,r2` must (entirely) lie on opposite sides of `J`. This is shown by considering a point `p` between `x1` and

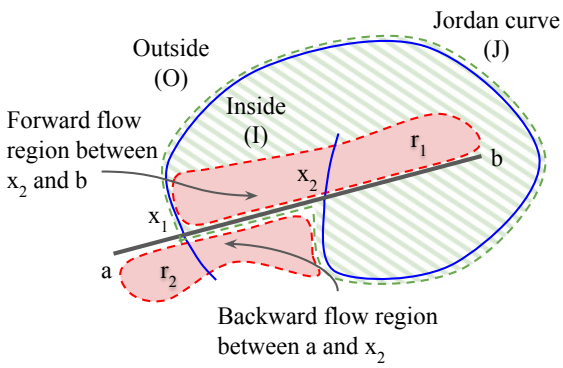


Figure 7. The constructed flow regions `r1`,`r2` for case Ⓐ. Note that `r1` is entirely inside, while `r2` is entirely outside `J`.

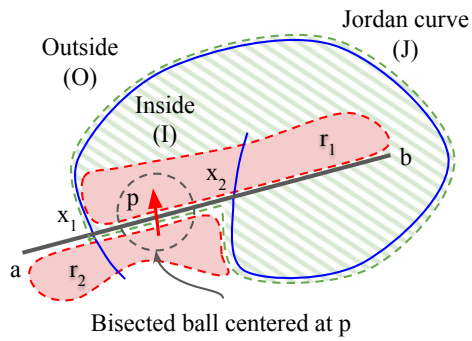


Figure 8. The bisected ball centered at `p` for case Ⓐ.

`x2` on the transversal segment. By construction, the forward flow from `p` is locally in `r1`, and the backward flow from `p` is locally in `r2`. Since `p` lies on the transversal segment and also in `J`, the Jordan curve bisects a small ball centered around `p`, such that one half of the ball is inside while the other is outside of `J`, see Fig. 8. Thus, the forward and backward flow from `p` must also locally lie on opposite sides of `J`. By path connectedness, regions `r1`,`r2` lie on opposite sides of `J`.

For the rest of this sketch, we consider the case where `r1` is entirely in `I`, while `r2` is in `O`. This corresponds to case Ⓐ as illustrated in Figures 7 and 8. The three properties can now be established simultaneously by analyzing the regions:

1. The forward (resp. backward) flow from points along the segment from `x1` to `x2` is locally in `r1` (resp. `r2`). Hence, the flow crosses along this segment from the outside to the inside.
2. The segment from `a` to `x1` consists entirely of limit points of `r2` so it lies in the closure of `O`. In fact, it lies entirely in `O` since it does not intersect `O`'s boundary `J`.
3. For time `t > t2` the flow locally lies in `r1`, and thus also locally in `I`. The flow stays in `I` for all `t > t2` by construction because it cannot intersect `J`. □

The monotonicity lemma required ≈760 lines of proof, not counting any sub-lemmas but including construction of the Jordan curve `J` and the flow regions `r1`,`r2`. Fortunately, subsequent proof steps can exploit symmetries in the problem. For example:

```
lemma flow0_transversal_segment_monotone_step_rev:
assumes "transversal_segment a b"
assumes "t1 ≤ t2" "{t1..t2} ⊆ existence_ivl0 x"
assumes x1: "flow0 x t1 ∈ {a<--<b}"
assumes x2: "flow0 x t2 ∈ {a<--<flow0 x t1}"
assumes "⋀t. t ∈ {t1<..<t2} ⟹ flow0 x t ∉ {a<--<b}"
assumes "t < t1" "t ∈ existence_ivl0 x"
shows "flow0 x t ∉ {a<--<flow0 x t1}"
```

This lemma utilizes two symmetries. First, the relative order of intersection at times `t1`,`t2` with the transversal segment is exchanged. Secondly, it draws a conclusion about the possible intersection at times `t < t1` instead of `t > t2`. Both symmetries exploit time reversal (Section 2.2.1). The former symmetry also uses the fact that transversal segments can be oriented in reverse. Together the proof of this lemma requires merely ≈30 lines of mostly boilerplate steps.

4.2.3 A Formal Proof (General Case)

The general case lemma is essentially identical except dropping the successive intersections assumption:

```
lemma flow0_transversal_segment_monotone:
assumes "transversal_segment a b"
assumes "t1 ≤ t2" "{t1..t2} ⊆ existence_ivl0 x"
assumes x1: "flow0 x t1 ∈ {a<--<b}"
assumes x2: "flow0 x t2 ∈ {flow0 x t1<--<b}"
assumes "t > t2" "t ∈ existence_ivl0 x"
shows "flow0 x t ∉ {a<--<flow0 x t2}"
```

Our formalization is similar to the textbook sketch. Like the textbook sketch, it uses a lemma (described in more detail in Section 5.5) showing that the number of intersections of the flow from `x` between times `t1`,`t2` and the transversal segment is finite. However, our proof avoids setting up a (tricky) induction. Briefly, by finiteness, there exists a maximum time `t` with `t1 ≤ t < t2` that is the last intersection time of the flow with the transversal segment before `t2`. By construction, there are no crossings between times `t`,`t2`. Let `y`,`x1`,`x2` be the points of intersection at times `t`,`t1`,`t2` of the flow with the transversal segment respectively. Firstly, note that `y ≠ x2`, otherwise the flow between `t` to `t2` is periodic and does not contain `x1`, contradiction. If `y < x2` according to the order of the transversal segment, then the conclusion follows by `flow0_transversal_segment_monotone_step` directly. Conversely, the case for `y > x2` is contradictory because then `flow0_transversal_segment_monotone_step_rev` would imply `x1 > y`, contradicting `x1 < x2`.

5 Formalization Techniques

In this section, we present, and reflect upon, certain design decisions in our formalization. These decisions helped to

keep the formalization in a maintainable state, i.e., without unnecessary duplication of code and with theorems phrased in the most canonical way. This is in order to make our formalization a library that can be used for developments going beyond just the proof of the Poincaré-Bendixson theorem.

We motivate the use of filters in general (Section 5.1) and show an application of filters to a recurring reasoning principle in dynamical systems (Section 5.2). We also explain how reasoning for symmetries in time is managed (Section 5.3), along with one approach to essentially changing coordinates in proofs (Section 5.4). Finally, we show examples of more general theorems formalized along the way that are not specific to the plane (Section 5.5).

5.1 Background: Filters

The `HOL-Analysis` libraries in Isabelle/HOL make heavy use of filters [15]. Filters are generalized quantifiers: a filter $\mathcal{F}$ is a predicate on predicates `𝓕 : ('a → bool) → bool`. Filter $\mathcal{F}$ applied to a predicate `P` is written as `∀F x in 𝓕. P x` instead of `𝓕 P` to hint at their intuitive reading as generalized quantifiers. For $\mathcal{F}$ to be a filter, it must satisfy the following properties (observe that an ordinary ∀-quantifier satisfies similar properties): it holds for the always `true` predicate and preserves conjunction as well as monotonicity.

```
lemma filter_properties:
 "(∀F x in 𝓕. True)" and
 "(∀F x in 𝓕. P x) ⟶ (∀F x in 𝓕. Q x) ⟶
    (∀F x in 𝓕. P x ∧ Q x)" and
 "(∀x. P x ⟶ Q x) ⟶ (∀F x in 𝓕. P x) ⟶
    (∀F x in 𝓕. Q x)"
```

Thus, properties that hold under a filter can be combined in a modular way, according to conjunction and monotonicity. For the dual one writes `∃F x in 𝓕. P x`.

```
lemma frequently:
 "(∃F x in 𝓕. P x) ⟷ ¬(∀F x in 𝓕. ¬ P x)"
```

For a detailed motivation and discussion of the role of filters in Isabelle/HOL we refer the reader to Hölzl *et al.* [15].

5.2 Frequent Choice

Recall from Section 3.1 that α- and ω-limit points and sets are defined in terms of (sub-)sequences, e.g, ω-limit points are witnessed by a sequence of times (tending to infinity), where the flow at those times tends to the ω-limit point.

Our formalization therefore requires a substantial amount of reasoning about (sub)sequences, mandating good reasoning infrastructure for them. In the spirit of `HOL-Analysis`, we use filters to make reasoning about such sequences as convenient and natural as possible.

A generalization of the introduction rule for ω-limit points can be stated with a binary predicate on natural and real numbers `P : nat → real → bool`. For the ω-limit point example, instantiate `P n t` with the predicate that denotes whether the flow at time `t` is `1/n` close to the ω-limit point in question. For an ω-limit point, `P` satisfies the property that for all `n` and `t0` one is able to choose a time `t > t0` such that `P n t` is true. More generally, under this assumption on `P`, one can choose a sequence of times `s` diverging to infinity such that `P i (s i)` is true for all elements of the sequence:

```
lemma frequently_at_top_elementary:
fixes P::"nat ⇒ real ⇒ bool"
assumes "∀n. ∀t0. ∃t>t0. P n t"
obtains s::"nat ⇒ real"
where "∀i. P i (s i)" "∀t. ∃N. ∀n≥N. s n ≥ t"
```

The proof of this lemma is not an issue for Isabelle/HOL: it is an easy application of the principle of dependent choice (an inductive application of the axiom of choice). Rather, we are concerned with trying to *use* the lemma. It contains alternating quantifiers in both assumption and conclusion, which makes reasoning tedious and cumbersome.

Filters provide an abstraction over the quantifier alternations. First, (part of) the assumption can be rewritten with the `at_top` filter, abstracting away one quantifier alternation:

```
lemma frequently_at_top:
 "(∃F t in at_top. P t) ⟷ (∀t0. ∃t>t0. P t)"
```

Second, the conclusion that the chosen sequence tends to infinity can be expressed more idiomatically and abstractly as a generalized limit. Recalling from Section 2.1, this removes another three quantifiers:

```
lemma filterlim_at_top_sequentially:
 "(s ⟶ ∞) ⟷ (∀t. ∃N. ∀n≥N. s n ≥ t)"
```

Combining these steps, a more idiomatic way of writing the `frequently_at_top_elementary` lemma is:

```
lemma frequently_at_top_realE:
fixes P::"nat ⇒ real ⇒ bool"
assumes "∀n. ∃F t in at_top. P n t"
obtains s::"nat ⇒ real"
where "⋀i. P i (s i)" "s ⟶ ∞"
```

This is just a reformulation of the elementary lemma (its proof is no more difficult) — but did we actually gain something from using filters? Yes indeed: the assumption of this new lemma is easier to establish and the conclusions are easier to continue reasoning with. For example, Isabelle/HOL's library contains many more lemmas to reason about the limit in the conclusion than about the specific combination of quantifiers. Moreover, filters can be combined directly, e.g, according to the following lemma.

```
lemma frequently_mp:
 "(∀F x in 𝓕. P x ⟶ Q x) ⟶ (∃F x in 𝓕. P x) ⟶
  (∃F x in 𝓕. Q x)"
```

This avoids the need to explicitly combine witnesses for the existential quantifiers behind, e.g., the `at_top` filter.

5.3 Reasoning Backward In Time

In the study of dynamical systems, many symmetries arise because the dynamics can be studied for forward or backward

time. Often, a symmetric case can be reduced to a known case by considering backward time, e.g., in Section 4.2.2.

This general idea was introduced in Section 2.2.1: the flow of an ODE is the same as the flow in backward time for the negated ODE. However, it is a technical challenge to actually exploit such symmetries and reuse results about forward time for backward time with as little additional proof effort or code duplication as possible. Our approach to exploiting the forward and backward symmetry in time is to make use of Isabelle/HOL's locale system [2] for modular reasoning.

A locale can be understood as a collection of theorems parameterized by a set of assumptions over them. The context in which we reason about ODEs, `c1_on_open f f' X`, is an example of a locale. Within a locale, one can introduce sublocale relationships which augments the current locale context with theorems from another locale. In our case, we introduce a sublocale relationship for the reverse ODE `-f`:

```
context c1_on_open f f' X begin
  sublocale rev: c1_on_open -f -f' X
```

This makes all theorems and definitions of the `c1_on_open` sublocale available (with a name prefix `rev`) for the reverse ODE. Now all theorems about the regular flow `flow0` as well as the flow for the reverse ODE `rev.flow0` are accessible. In the notation of Section 2.2.1, `rev.flow0` corresponds to $\texttt{flow0}_{(-f)}$. This allows us to (easily) prove, once-and-for-all, rules relating the flow, existence interval, transversal segment, and limit points for forward and backward time:

```
lemma rev_locale_properties:
 "rev.flow0 y t = flow0 y (-t)"
 "t ∈ rev.existence_ivl0 y ⟷ -t ∈ existence_ivl0 y"
 "rev.transversal_segment = transversal_segment"
 "rev.ω_limit_point = α_limit_point"
```

As an example application, the time-reversed version of lemma `ω_limit_crossings` from ② in Section 4.1 that is automatically generated in the `rev` sublocale is the following:

```
lemma α_limit_crossings_raw:
assumes "rev.transversal_segment a b"
assumes "rev.ω_limit_point x p"
assumes "p ∈ {a<--<b}"
obtains s where
 "s ⟶ ∞" "(rev.flow0 x ∘ s) ⟶ p"
 "∀n. rev.flow0 x (s n) ∈ {a<--<b}"
```

Rewriting time reversed concepts to their original counterparts yields the corresponding lemma for α-limit points:

```
lemma α_limit_crossings:
assumes "transversal_segment a b"
assumes "α_limit_point x p"
assumes "p ∈ {a<--<b}"
obtains s where
 "s ⟶ -∞" "(flow0 x ∘ s) ⟶ p"
 "∀n. flow0 x (s n) ∈ {a<--<b}"
```

Notably, this lemma is obtained by simple rewriting, hence achieving the goal of avoiding duplicate proofs for time symmetric lemmas.

5.4 Line Segments

Proper abstraction is the key to properly engineered proofs and mathematics. But one must be careful not to abstract away too much information that must otherwise be recovered tediously. Our example for this is line segments, defined as the set of convex combinations of the endpoints.

```
lemma closed_segment_def: "{a--b} =
  {(1 - u) *R a + u *R b | u::real. 0 ≤ u ∧ u ≤ 1}"
```

This representation as a set hides information that would have been intuitively useful when reasoning about segments: namely that the points on the segment can be ordered. In our formalization (Section 4), the fact that two points `x` and `y` are ordered on the line segment `{a--b}` such that `x` is closer to `a` is expressed like so (with `<--<` instead of `--` if `x ≠ y`):

```
assume "x ∈ {a--b}" "y ∈ {x--b}"
```

This correctly expresses the ordering, but reasoning with it is hard: essentially all proofs about it required expanding definitions down to reasoning about the convex combinations making up the line segments. This often made our proofs about orderings on line segments very tedious.

We therefore introduced a layer of abstraction between the set `{a--b}` and its underlying convex combinations. Namely, we introduce $\texttt{\{a--b\}}_{\texttt{u}}$ and parameterize the (whole) line through `a` and `b` with a real number `u`.

```
lemma line_def: "{a--b}u = a + u *R (b - a)"
```

The parameterization is chosen such that the line segment between `a` and `b` is the image of the unit interval. Note that `line` and theorems about it can be reused for different kinds of segments: (half) open line segments can simply be represented as the image of `line` of (half) open (unit) intervals.

```
lemma closed_segment_line: "{a--b} = {a--b}`{0..1}"
lemma open_segment_line:
 "a ≠ b ⟹ {a<--<b} = {a--b}`{0<..<1}"
```

With this notion of `line`, a better way to express the ordering of `x` and `y` is by directly referring to the order on the underlying unit interval:

```
assume "x = {a--b}i" "y = {a--b}j"
  "0 ≤ i ∧ i ≤ j ∧ j ≤ 1"
```

The more abstract notion of membership in line segments `x ∈ {a--b}` is used in lemma statements. As a first step in proofs, we change coordinates by obtaining the parameter `i` on the unit interval such that $\texttt{x = \{a--b\}}_{\texttt{i}}$. After that, reasoning about the order on segments is reduced to reasoning about the order on the unit interval. There Isabelle/HOL's arithmetic tactics apply with ease and have simplified many previously cumbersome proofs about the order of points on open, closed, and combinations of open and closed segments.

5.5 Generalizations

To promote reuse of the formalization, we formalized all results as generally as possible. For example, this is the (planar)

lemma that in finite time, the flow can intersect a transversal segment (a line segment in $\mathbb{R}^2$) only finitely many times:

```
lemma flow_transversal_segment_finite_intersections:
assumes "transversal_segment a b"
assumes "t1 ≤ t2" "{t1..t2} ⊆ existence_ivl0 x"
shows "finite {s ∈ {t1..t2}. flow0 x s ∈ {a--b}}"
```

This theorem is actually true (and proved) in a much more general setting: a flow in $\mathbb{R}^n$ intersects a differentiable surface (represented as the intersection of a closed set S and the zeroes of a differentiable function s) only finitely many times:

```
lemma flow_transversal_surface_finite_intersections:
fixes s::"'a ⇒ 'b::real_normed_vector"
 and Ds::"'a ⇒ 'a ⇒L 'b"
assumes "closed S"
assumes "⋀x. (s has_derivative (Ds x)) (at x)"
assumes "⋀x. x ∈ S ⟹ s x = 0 ⟹ Ds x (f x) ≠ 0"
assumes "t1 ≤ t2" "{t1..t2} ⊆ existence_ivl0 x"
shows
 "finite {t ∈ {t1..t2}. flow0 x t ∈ {x ∈ S. s x = 0}}"
```

The formalization of limit sets and periodic orbits (Section 3.1) was also done in a more general setting. It is formalized in the locale of locally Lipschitz continuous RHS for the ODE in finite dimensional vector spaces, rather than just for continuously differentiable RHS on the Euclidean plane.

6 Existence of Limit Cycles

This section uses the Poincaré-Bendixson theorem to formalize the existence of *limit cycles* on two examples. A limit cycle is a periodic orbit that is the α- or ω-limit set of a point *not* already contained in the cycle [6].[9] Formally:

```
lemma limit_cycle_def: "limit_cycle y ⟷
  periodic_orbit y ∧
  (∃x. x ∉ flow0 y ‘ UNIV ∧
  (flow0 y ‘ UNIV = ω_limit_set x ∨
    flow0 y ‘ UNIV = α_limit_set x))"
```

The existence of limit cycles can be established using the following corollary of poincare_bendixson:

```
corollary poincare_bendixson_limit_cycle:
assumes "compact K" "K ⊆ X" "0 ∉ f ‘ K"
assumes "positively_invariant K"
assumes "x ∈ K"
assumes "t ∈ existence_ivl0 x ∩ {..0}"
assumes "flow0 x t ∉ K"
obtains y where
 "limit_cycle y"
 "flow0 y ‘ UNIV ⊆ K"
```

The proof of this corollary closely follows how it is used. To use the corollary, one supplies a set K and a point x. The (positively) invariant compact set K acts as a *trapping region* because its (positive) invariance guarantees that the flow forward from any point in K, including x, is trapped forward in K. From these properties of K, poincare_bendixson guarantees that the ω-limit set of x is a periodic orbit (from a point y) that is contained within K.

The last two assumptions, t ∈ existence_ivl0 x ∩ {..0} and flow0 x t ∉ K say that x can be flowed backward for some time t in its negative existence interval to exit K. This guarantees that x is not in the periodic orbit from y because the periodic orbit is trapped for forward *and backward* time in K. Thus, the periodic orbit from y is a limit cycle. These last two assumptions are stated differently compared to the literature [6, Thm. 1.179]. They enable the use of verified ODE reachability analysis [17] in the examples below.

6.1 Circle Example

The main ideas are first illustrated on a textbook example [6, Chap. 1.9] visualized in Fig. 9 (Left), where the limit cycle is the unit circle. The prefix c below (e.g., c.flow0) refers to the locale instance for this example. The (compact) *annular* trapping region cK is chosen to be the set of points between the circles of radius 2 and radius $\frac{1}{2}$. This choice of cK excludes the equilibrium point at the origin and satisfies 0 ∉ f ‘ cK.

The main challenge is proving positively_invariant cK. From Fig. 9, this is intuitively true because the arrows always point "inwards" on the boundary of cK. To prove this, we formalized a comparison principle [33, §9.IX] and, as corollaries, variations of *barrier certificate* principles [31] that can be used to establish (positive) invariance. Technical details of this formalization are omitted as it is not the focus of this paper. The use of these principles reduces invariance to (real) arithmetic goals, which are discharged using Isabelle/HOL's builtin linear and sum-of-squares arithmetic.

For example, to show positive invariance for cK, one step in the proof is to show that the outer disc of radius 2 is positively invariant:

```
lemma positively_invariant_outer:
 "c.positively_invariant (cball (0, 0) 2)"
```

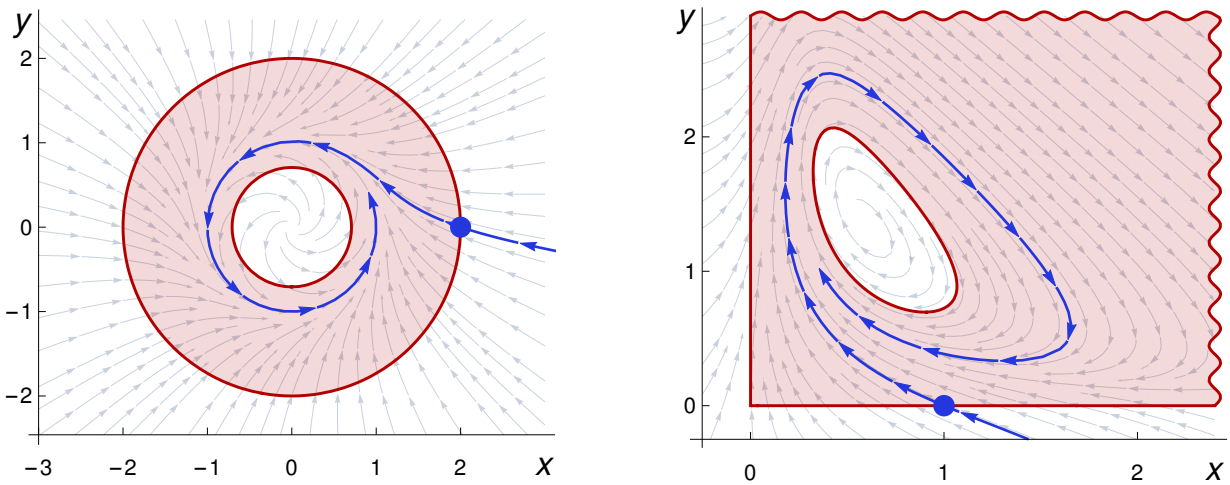
An application of the barrier certificate principle reduces this to a question of real arithmetic that is solved automatically by the sum-of-squares proof method sos:

```
lemma c_arith:
 "2 * (-y + x * (1 - x^2 - y^2)) * x +
  2 * ( x + y * (1 - x^2 - y^2)) * y
  ≤ (-(2 * x^2) - 2 * y^2) * (x^2 + y^2 - 4)"
```

The point x is chosen to be (2,0), and Isabelle/HOL's verified ODE reachability analysis [17] tool is used to compute bounds on the backward flow from x. For time t=-0.01, the flow is proved to lie in an enclosing box:

```
lemma c_reachable:
 "-0.01 ∈ c.existence_ivl0 (2, 0)"
 "c.flow0 (2, 0) (-0.01) ∈
    {(2.06,-0.021)..(2.07,-0.02)}"
```

[9] Readers may be familiar with the *less general* definition of limit cycles as isolated periodic orbits, see [6, Def. 1.178].



(Left) $\dot{x} = -y + x(1 - x^2 - y^2), \dot{y} = x + y(1 - x^2 - y^2)$

(Right) $\dot{x} = -x + 0.08y + x^2y, \dot{y} = 0.6 - 0.08y - x^2y$

Figure 9. The compact trapping region `K` (in shaded red) together with initial point `x` (in blue) whose ω-limit set is a limit cycle, respectively for the circle example (Left) and the glycolysis example (Right). The trapping region boundaries are drawn in solid red. For the glycolysis example (Right), the wavy red boundary indicates a truncated plot of the trapping region.

The box `{(2.06,-0.021)..(2.07,-0.02)}` is bounded away from `cK`, so applying `poincare_bendixson_limit_cycle` yields the existence of a limit cycle for this example:

```
theorem c_has_limit_cycle:
obtains y where
 "c.limit_cycle y" "c.flow0 y ‘ UNIV ⊆ cK"
```

6.2 Glycolysis Example

For the second example, we show that a limit cycle exists for Sel'kov's model of glycolysis for the parameter values $a = 0.08$ and $b = 0.6$ [30, 32] (see Fig. 1). The proof largely follows the same ideas as the textbook example above and the prefix `g` is used below to refer to the locale instance for this example. The trapping region `gK` and point `x` are visualized in Fig. 9 (Right). The most involved part of this example is the construction of the trapping region and proving its positive invariance. We mostly follow the construction in [32, Example 7.3.2], except for the inner excluded region, see Fig. 9, which is described by the degree 4 polynomial `p1` below found using sum-of-squares programming:

```
lemma p1_def:
 "p1 (x, y) = -21/34 - 69*x/38 + 19*x^2/15 -
    9*x^3/28 - 6*x^4/43 + 14*y/29 + 31*x*y/21 +
    182*x^2*y/47 - 35*x^3*y/16 - 3*y^2/17 -
    2*x*y^2/9 - 31*x^2*y^2/20 + y^3/102 + x*y^3/59"
```

Proving the required properties for `gK` is complicated by the presence of `p1` in its description. Here, the assumptions `compact gK` and `0 ∉ f ' gK` are discharged by linear and sum-of-squares arithmetic. However, proving positive invariance for `gK` results in the following difficult real arithmetic goal (related to `p1`), where `p1' (x, y)` is the total derivative (more precisely, the Jacobian) of `p1` at `(x, y)`:

```
lemma g_arith:
 "(- (27 / 25) - x^2 + 2 * x * y) * p1 (x, y) -
  p1' (x, y) (-x + 0.08 * y + x^2 * y,
              0.6 - 0.08 * y - x^2 * y) ≥ 0"
if xb: "0 ≤ x" "x ≤ 8.24" and yb: "0 ≤ y" "y ≤ 7.51"
```

This problem is too challenging for Isabelle/HOL's sum-of-squares solver: even after 5 minutes, it does not return with a solution. Fortunately, the compactness of the bounds (`xb`, `yb`) allows us to prove the arithmetic goal numerically using affine arithmetic [16] and a branch-and-bound technique.

The final theorem (some proof steps omitted) shows that a limit cycle exists within the trapping region `gK`, and thus that Sel'kov's model exhibits limiting periodic behavior:

```
theorem g_has_limit_cycle:
obtains y where
 "g.limit_cycle y" "g.flow0 y ‘ UNIV ⊆ gK"
```

We note that Sel'kov's model actually exhibits limit cycles for all parameter values a, b that satisfy a particular relationship [32, Example 7.3.3]. The use of concrete parameters a, b in this example is to illustrate our approach, including the use of verified ODE reachability analysis and various forms of real arithmetic proofs in Isabelle/HOL.

7 Related Work

The Poincaré-Bendixson theorem has a rich history and is reported in several standard textbooks in ordinary differential equations and dynamical systems [6, 7, 9, 25, 28, 33, 34]. We are not aware of any other existing (or ongoing) effort in formalizing this theorem. The Jordan curve theorem was

formalized by Hales [12] and independently by Harrison [13] in HOL light. Harrison's proof was ported to Isabelle/HOL by Paulson and subsequently used in our work. Some other porting efforts by Paulson are briefly mentioned in a different context [1]. The library for analysis in Isabelle/HOL, `HOL-Analysis`, has its origins in a formalization of nonstandard real analysis by Fleuriot and Paulson [10], and a formalization of multivariate analysis by Harrison [13] in HOL Light. Hölzl *et al.* [15] describe how the formalization was developed further to profit from Isabelle/HOL's type class system and center the treatment of limits around filters.

Our formalization builds substantially on Immler *et al.* [18, 19, 21]'s formalization of ordinary differential equations in Isabelle/HOL. This library has also been used to formalize reasoning for hybrid systems (and hybrid games) [5, 11, 26, 35] that combine discrete, adversarial, and continuous dynamics, the latter of which is specified by ordinary differential equations. Whereas our formalization focuses on dynamical systems theory, some of the results, e.g., ω-limit sets could be of useful in formalizing reasoning principles relevant to the study of hybrid systems.

Cohen and Rouhling [8] formalized LaSalle's invariance principle in the Coq proof assistant. This formalization was later used by Rouhling to formalize the correctness of a controller for the inverted pendulum [29]. LaSalle's principle uses properties of the ω-limit set; all of the required properties have been formalized in our work (Section 3). The most notable difference between our formalization and theirs is the hypothesis: Cohen and Rouhling [8] hypothesize the *global* existence of a solution to the differential equations, which obviates the need to manage any kind of existence reasoning for solutions. In contrast, ours builds on the true solution obtained via Picard iteration [18, 21] — solutions of differential equations *do not*, in general, exist globally but only on an open existence interval [6]. The solution obtained by Picard iteration is also formalized in Coq by Makarov and Spitters [22] in a constructive setting.

8 Conclusion and Future Work

The Poincaré-Bendixson theorem serves as an interesting watershed for the maturity of a proof assistant's mathematical analysis libraries. It also provides a challenging case study involving the formalization of (seemingly) straightforward geometric arguments. Although our formalization of the theorem is in Isabelle/HOL, we believe that our proof, especially the flow region construction from Section 4.2, can be used as a blueprint in any other proof assistant with the requisite analysis libraries. On the other hand, our formalization directly reasons about the given differential equations. In some textbook proofs [6, 9], the rectification theorem [6, Lem. 1.120] is used to first (locally) place the differential equations into a particularly nice geometric form, which may, e.g., simplify the proof of the monotonicity lemma. It would be interesting to compare the proof effort between these two approaches, e.g., by exploring if rectification-type arguments can be formalized conveniently using sublocale relationships similarly to Section 5.3.

For future work, the Poincaré-Bendixson theorem is just one of many tools that can be used to analyze (planar) dynamical systems. Related tools that are ripe for formalization include: *i)* Liénard's theorem [25, Chap. 3.8], which shows the existence and uniqueness of a *stable* limit cycle for certain planar systems. Such stability properties are of interest, e.g., in control theory and in the study of population dynamics, as they guarantee the oscillatory behavior of such systems even under minor disturbances. *ii)* Dulac's (or Bendixson's) criterion [6, Prop. 1.195], that can be used to establish *non-existence* of periodic orbits. As Section 6 demonstrates, these tools can be used to formally analyze systems of interest. Developing a library of such tools in Isabelle/HOL could also yield, as a byproduct, an interesting library of geometric reasoning techniques. Finally, the Poincaré-Bendixson theorem is also true for continuous dynamics defined on the cylinder and two-sphere [25, 34]; formalizing these variations in Isabelle/HOL would require, e.g., appropriate generalizations of the Jordan curve theorem and the theory of ordinary differential equations for those surfaces.

Acknowledgments

We thank Brandon Bohrer and the anonymous reviewers for their helpful feedback on this paper.

References

[1] Mohammad Abdulaziz and Lawrence C. Paulson. 2019. An Isabelle/HOL Formalisation of Green's Theorem. *J. Autom. Reasoning* 63, 3 (2019), 763–786. https://doi.org/10.1007/s10817-018-9495-z

[2] Clemens Ballarin. 2014. Locales: A Module System for Mathematical Theories. *J. Autom. Reasoning* 52, 2 (2014), 123–153. https://doi.org/10.1007/s10817-013-9284-7

[3] Anatole Beck. 1974. *Continuous Flows in the Plane.* Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-65548-7

[4] Ivar Bendixson. 1901. Sur les courbes définies par des équations différentielles. *Acta Math.* 24, 1 (1901), 1–88. https://doi.org/10.1007/BF02403068

[5] Brandon Bohrer, Vincent Rahli, Ivana Vukotic, Marcus Völp, and André Platzer. 2017. Formally Verified Differential Dynamic Logic. In *CPP*, Yves Bertot and Viktor Vafeiadis (Eds.). ACM, 208–221. https://doi.org/10.1145/3018610.3018616

[6] Carmen Chicone. 2006. *Ordinary Differential Equations with Applications* (second ed.). Springer, New York. https://doi.org/10.1007/0-387-35794-7

[7] Earl A. Coddington and Norman Levinson. 1955. *Theory of Ordinary Differential Equations.* McGraw-Hill Book Company, Inc., New York-Toronto-London.

[8] Cyril Cohen and Damien Rouhling. 2017. A Formal Proof in Coq of LaSalle's Invariance Principle. In *ITP (LNCS)*, Mauricio Ayala-Rincón and César A. Muñoz (Eds.), Vol. 10499. Springer, 148–163. https://doi.org/10.1007/978-3-319-66107-0_10

[9] Freddy Dumortier, Jaume Llibre, and Joan C. Artés. 2006. *Qualitative Theory of Planar Differential Systems.* Springer, Berlin.

[10] Jacques D. Fleuriot and Lawrence C. Paulson. 2000. Mechanizing Nonstandard Real Analysis. *LMS Journal of Computation and Mathematics* 3 (2000), 140–190. https://doi.org/10.1112/S1461157000000267

[11] Simon Foster. 2019. Hybrid Relations in Isabelle/UTP. In *UTP (LNCS)*, Pedro Ribeiro and Augusto Sampaio (Eds.), Vol. 11885. Springer, 130–153. https://doi.org/10.1007/978-3-030-31038-7_7

[12] Thomas C. Hales. 2007. The Jordan Curve Theorem, Formally and Informally. *The American Mathematical Monthly* 114, 10 (2007), 882–894. http://www.jstor.org/stable/27642361

[13] John Harrison. 2013. The HOL Light Theory of Euclidean Space. *J. Autom. Reasoning* 50, 2 (2013), 173–190. https://doi.org/10.1007/s10817-012-9250-9

[14] Morris W. Hirsch. 1984. The Dynamical Systems Approach to Differential Equations. *Bull. Amer. Math. Soc. (N.S.)* 11, 1 (1984), 1–64. https://doi.org/10.1090/S0273-0979-1984-15236-4

[15] Johannes Hölzl, Fabian Immler, and Brian Huffman. 2013. Type Classes and Filters for Mathematical Analysis in Isabelle/HOL. In *ITP (LNCS)*, Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie (Eds.), Vol. 7998. Springer, 279–294. https://doi.org/10.1007/978-3-642-39634-2_21

[16] Fabian Immler. 2014. Affine Arithmetic. *Archive of Formal Proofs* (Feb. 2014). http://isa-afp.org/entries/Affine_Arithmetic.html, Formal proof development.

[17] Fabian Immler. 2015. Verified Reachability Analysis of Continuous Systems. In *TACAS (LNCS)*, Christel Baier and Cesare Tinelli (Eds.), Vol. 9035. Springer, 37–51. https://doi.org/10.1007/978-3-662-46681-0_3

[18] Fabian Immler and Johannes Hölzl. 2012. Numerical Analysis of Ordinary Differential Equations in Isabelle/HOL. In *ITP (LNCS)*, Lennart Beringer and Amy P. Felty (Eds.), Vol. 7406. Springer, 377–392. https://doi.org/10.1007/978-3-642-32347-8_26

[19] Fabian Immler and Johannes Hölzl. 2019. Ordinary Differential Equations. *Archive of Formal Proofs* (2019). http://isa-afp.org/entries/Ordinary_Differential_Equations.html, Formal proof development.

[20] Fabian Immler and Yong Kiam Tan. 2019. The Poincaré-Bendixson Theorem. *Archive of Formal Proofs* (Dec. 2019). http://isa-afp.org/entries/Poincare_Bendixson.html, Formal proof development.

[21] Fabian Immler and Christoph Traut. 2019. The Flow of ODEs: Formalization of Variational Equation and Poincaré Map. *J. Autom. Reasoning* 62, 2 (2019), 215–236. https://doi.org/10.1007/s10817-018-9449-5

[22] Evgeny Makarov and Bas Spitters. 2013. The Picard Algorithm for Ordinary Differential Equations in Coq. In *ITP (LNCS)*, Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie (Eds.), Vol. 7998. Springer, 463–468. https://doi.org/10.1007/978-3-642-39634-2_34

[23] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. 2002. *Isabelle/HOL - A Proof Assistant for Higher-Order Logic*. LNCS, Vol. 2283. Springer. https://doi.org/10.1007/3-540-45949-9

[24] Lawrence C. Paulson. 1989. The Foundation of a Generic Theorem Prover. *J. Autom. Reasoning* 5, 3 (1989), 363–397. https://doi.org/10.1007/BF00248324

[25] Lawrence Perko. 1991. *Differential Equations and Dynamical Systems*. Vol. 7. Springer, New York. https://doi.org/10.1007/978-1-4684-0392-3

[26] André Platzer. 2019. Uniform Substitution At One Fell Swoop. In *CADE (LNCS)*, Pascal Fontaine (Ed.), Vol. 11716. Springer, 425–441. https://doi.org/10.1007/978-3-030-29436-6_25

[27] Henri Poincaré. 1881. Mémoire sur les courbes définies par une équation différentielle. *J. Math. Pures Appl.* (1881).

[28] Clark Robinson. 1999. *Dynamical Systems: Stability, Symbolic Dynamics, and Chaos* (second ed.). CRC Press, Boca Raton, FL.

[29] Damien Rouhling. 2018. A Formal Proof in Coq of a Control Function for the Inverted Pendulum. In *CPP*, June Andronick and Amy P. Felty (Eds.). ACM, 28–41. https://doi.org/10.1145/3167101

[30] Evgeni E. Sel'kov. 1968. Self-Oscillations in Glycolysis. A Simple Kinetic Model. *European Journal of Biochemistry* 4, 1 (1968), 79–86. https://doi.org/10.1111/j.1432-1033.1968.tb00175.x

[31] Andrew Sogokon, Khalil Ghorbal, Yong Kiam Tan, and André Platzer. 2018. Vector Barrier Certificates and Comparison Systems. In *FM (LNCS)*, Klaus Havelund, Jan Peleska, Bill Roscoe, and Erik P. de Vink (Eds.), Vol. 10951. Springer, 418–437. https://doi.org/10.1007/978-3-319-95582-7_25

[32] Steven H. Strogatz. 2015. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering* (second ed.). Westview Press, Boulder, CO.

[33] Wolfgang Walter. 1998. *Ordinary Differential Equations*. Springer, New York. https://doi.org/10.1007/978-1-4612-0601-9

[34] Stephen Wiggins. 2003. *Introduction to Applied Nonlinear Dynamical Systems and Chaos* (second ed.). Vol. 2. Springer, New York. https://doi.org/10.1007/b97481

[35] Jonathan Julián Huerta y Munive and Georg Struth. 2018. Verifying Hybrid Systems with Modal Kleene Algebra. In *RAMiCS (LNCS)*, Jules Desharnais, Walter Guttmann, and Stef Joosten (Eds.), Vol. 11194. Springer, 225–243. https://doi.org/10.1007/978-3-030-02149-8_14