EPiC Series in Computing

Volume 108, 2025, Pages 39–70

Proceedings of 12th Int. Workshop on Applied Verification for Continuous and Hybrid Systems

ARCH-COMP25 Category Report: Continuous and Hybrid Systems with Nonlinear Dynamics

Luca Geretti[1], Julien Alexandre dit Sandretto[2], Matthias Althoff[3], Luis Benet[4], Pieter Collins[5], Marcelo Forets[6], Stefan Mitsch[7], Ismail Patel[7], Maximilian Perschl[3], Christian Schilling[8], and Joris Tillet[2]

[1] Department of Computer Science, University of Verona, Verona, Italy
luca.geretti@univr.it
[2] ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France
julien.alexandre-dit-sandretto@ensta-paris.fr,joris.tillet@ensta-paris.fr
[3] Technische Universität München, Munich, Germany
althoff@tum.de,max.perschl@tum.de
[4] Instituto de Ciencias Físicas, Universidad Nacional Autónoma de México (UNAM), México
benet@icf.unam.mx
[5] Department of Advanced Computer Sciences, Maastricht University, Maastricht, The Netherlands
pieter.collins@maastrichtuniversity.nl
[6] Universidad de la República, Montevideo, Uruguay
mforets@gmail.com
[7] School of Computing, DePaul University, Chicago, IL, USA
smitsch@depaul.edu,ipatel31@depaul.edu
[8] Aalborg University, Aalborg, Denmark
christianms@cs.aau.dk

Abstract

We present the results of a friendly competition for formal verification of continuous and hybrid systems with nonlinear continuous dynamics. The friendly competition took place as part of the workshop Applied Verification for Continuous and Hybrid Systems (ARCH) in 2025. This year, 5 tools participated: Ariadne, CORA, DynIbex, JuliaReach and KeYmaera X (in alphabetic order). These tools are applied to solve reachability analysis problems on seven benchmark problems, three of them featuring some aspects of hybrid dynamics. We do not rank the tools based on the results, but show the current status and discover the potential advantages of different tools.

1 Introduction

Disclaimer The presented report of the ARCH-COMP friendly competition for *continuous and hybrid systems with nonlinear dynamics* aims at providing a landscape of the current capabilities of verification tools. We would like to stress that each tool has unique strengths—though not all of their features can be highlighted within a single report. To reach a consensus in what benchmarks are used, some compromises had to be made so that some tools may benefit more from the presented choice than others. The obtained results have been verified by an independent repeatability evaluation. To establish further trustworthiness of the results, the code with which the results have been obtained is publicly available as Docker [20] containers at gitlab.com/goranf/ARCH-COMP.

In this report, we summarize the results of the ninth ARCH-COMP [1] friendly competition on the reachability analysis of continuous and hybrid systems with nonlinear dynamics. Given a system defined by a nonlinear Ordinary differential equation (ODE) $\dot{\vec{x}} = f(\vec{x}, t)$ along with an initial condition $\vec{x} \in X_0$, we apply the participating tools to prove properties of the state reachable set in a bounded time horizon. The techniques for solving such a problem are usually very sensitive to not only the nonlinearity of the dynamics but also the size of the initial set. This is also one of the main reasons why most of the tools require quite a lot of computational parameters.

In this report, 5 tools, namely Ariadne, CORA, DynIbex, JuliaReach and KeYmaera X, participated in solving problems defined on five continuous and two hybrid benchmarks. The continuous benchmarks are the Traffic scenario, the Robertson chemical reaction system, the Coupled Van der Pol oscillator, the Laub-Loomis model of enzymatic activities and the Transient Stability analysis of Power Systems. The hybrid benchmarks model a Lotka-Volterra predator-prey system with a Tangential Crossing, and a Space Rendezvous system.

The benchmarks were selected based on discussions between the tool authors, with a preference on keeping a significant set of the benchmarks from the previous year. It is apparent that they come from very different domains and aim at identifying issues specific to nonlinear dynamics, possibly with the addition of hybrid behavior. In particular, with the Power Systems benchmark we introduced differential-algebraic dynamics for the first time.

2 Participating Tools

Ariadne. (Luca Geretti, Pieter Collins) *Ariadne* [28, 19] is a library based on *Computable Analysis* [53] that uses a rigorous numerical approach to all its algebraic, geometric and logical operations. In particular, it performs numerical rounding control of all external and internal operations, in order to enforce conservative interpretation of input specification and guarantee formal correctness of the computed output. It focuses on nonlinear systems, both continuous and hybrid, supporting differential and algebraic relations, with a focus on compositionality [23, 24]. It has been mainly applied to the verification of robotic tasks [34, 25, 26]. The library is written in modern C++ with an optional Python interface. The official site for Ariadne is `https://www.ariadne-cps.org`.

CORA. (Matthias Althoff, Maximilian Perschl) The Continuous Reachability Analyzer (CORA) [8, 9] is a MATLAB-based toolbox designed for the formal verification of cyber-physical systems through reachability analysis. It offers a comprehensive suite of tools for modeling and analyzing various system dynamics, including linear, nonlinear, and hybrid systems. CORA supports both continuous and discrete-time systems, accommodating uncertainties in system inputs and parameters. These uncertainties are captured by a diverse range of set representations such as intervals, zonotopes, Taylor models, and polytopes. Additionally, CORA provides functionalities for the formal verification of neural networks as well as data-driven system identification with reachset conformance. Various converters are implemented to easily model a system in CORA such as the well-established SpaceEx format for dynamic systems and ONNX format for neural networks. CORA ensures the seamless integration of different reachability algorithms without code modifications and aims for a user-friendly experience through automatic parameter tuning, making it a versatile tool for researchers and engineers in the field of cyber-physical systems. CORA is available at cora.in.tum.de.

DynIbex. (Julien Alexandre dit Sandretto, Joris Tillet) A library merging interval constraint satisfaction problem algorithms and guaranteed numerical integration methods based on Runge-Kutta numerical schemes implemented with affine arithmetic. This library is able to solve ordinary differential equations [3] and algebraic differential equations of index 1 [4], combined with numerical constraints on state variables and reachable tubes. It produces **sound results** taking into account round-off errors in floating-point computations and truncation errors generated by numerical integration methods [43]. Moreover, constraint satisfaction problem algorithms offer a convenient approach to check properties on reachable tubes as explained in [5]. This library implements in a very generic way validated numerical integration methods based on Runge-Kutta methods without many optimizations. Indeed, the computation of the local truncation error, for each method, depends only on the coefficients of Runge-Kutta methods and their order. DynIbex is freely available at `http://perso.ensta-paristech.fr/~chapoutot/dynibex/`. Figures have been produced with VIBes library [29] which is available at `http://enstabretagnerobotics.github.io/VIBES/`.

JuliaReach. (Luis Benet, Marcelo Forets, Christian Schilling) JuliaReach [21] is an open-source software suite for reachability computations of dynamical systems, written in the Julia language and available at `http://github.com/JuliaReach`. Linear, nonlinear, and hybrid problems are modeled and solved using the library ReachabilityAnalysis.jl, which can be used interactively, for example in Jupyter notebooks. Our implementation of the Taylor-model based solvers (`TMJets21a` and `TMJets21b`), which are implemented in TaylorModels.jl [18], incorporates the packages TaylorSeries.jl [15, 16] and TaylorIntegration.jl [44], and the IntervalArithmetic.jl [17] package for interval methods. The algorithms applied in this report first compute a non-validated integration using a Taylor model of order n_T. The coefficients of that series are polynomials of order n_Q in the variables that denote small deviations of the initial conditions. We obtain a time step from the last two coefficients of this time series. In order to validate the integration step, we compute a second integration using intervals as coefficients of the polynomials in time, and we obtain a bound for the integration using a Lagrange-like remainder. The remainder is used to check the contraction of a Picard iteration. If the combination of the time step and the remainder do not satisfy the contraction, we iteratively enlarge the remainder or possibly shrink the time step. Finally, we evaluate the initial Taylor series with the valid

remainder at the time step for which the contraction has been proved, which is also evaluated in the initial set to yield an over-approximation. The approach is (numerically) sound due to rigorous interval bounds in the Taylor approximation. Discrete transitions for hybrid systems and Taylor-model approximations are handled using the set library LazySets.jl [30].

KeYmaera X. (Stefan Mitsch, Ismail Patel) KeYmaera X [32] is a theorem prover for the hybrid systems logic differential dynamic logic (dL). It implements the uniform substitution calculus of dL [46]. A comparison of the internal reasoning principles in the KeYmaera family of provers with a discussion of their relative benefits and drawbacks is in [42], and model structuring and proof management on top of uniform substitution is discussed in [39]. KeYmaera X supports systems with nondeterministic discrete jumps, nonlinear differential equations, nondeterministic inputs, and allows defining functions implicitly through their characterizing differential equations [33]. It provides invariant construction and proving techniques for differential equations [50, 47], stability verification techniques for switched systems [51], and tactics for certifying Taylor models, inspired by [35]. Unlike numerical hybrid systems reachability analysis tools, KeYmaera X also supports unbounded initial sets and unbounded time analysis. Proofs in KeYmaera X can be conducted interactively [40], steered with tactics [31], or attempted fully automatic.

3 Benchmarks

All benchmarks are identified with a four letter code and the last two digits of the year it was most recently updated. A total of seven benchmarks is available this year: TRAF22, ROBE25, CVDP23 and LALO20 for continuous systems, and LOVO25, SPRE22 and TSPS25 for hybrid systems. For each benchmark, the Model subsection defines the dynamics, Analysis specifies how to run the model, Evaluation defines the metrics used to compare tools, and Results discusses the actual results, in particular explaining the meaningful settings required for each tool.

3.1 Traffic scenario benchmark (TRAF22)

The avoidance of collisions in traffic scenarios is of utmost interest in the development of motion planners for autonomous driving. Recently [37], a workflow for the automated generation of verification tasks has been proposed based on an extraction of traffic scenario benchmarks from the CommonRoad framework [10].

3.1.1 Model

The nonlinear continuous-time dynamics are represented by a kinematic single-track model [37, Eq. (1)]:

$$\begin{cases} \dot{\delta} = u_1 + w_1 \\ \dot{\psi} = \dfrac{v}{l_{wb}} \tan\delta \\ \dot{v} = u_2 + w_2 \\ \dot{s}_x = v\cos\psi \\ \dot{s}_y = v\sin\psi, \end{cases}$$

where the state vector $x \in \mathbb{R}^5$ consists of the steering angle δ, the vehicle heading ψ, the vehicle velocity v, and the positions s_x, s_y of the vehicle along the x-axis and y-axis. The control inputs u_1, u_2 represent the steering angle and acceleration, respectively. Additionally, model uncertainties and disturbances affecting the vehicle are modeled by the disturbances w_1, w_2. In order to follow a reference trajectory $x_{ref} \in \mathbb{R}^5$, we apply a feedback controller of the form [37, Eq. (2)]

$$u_{fb}(\widehat{x}) = u_{ref} + K(\widehat{x} - x_{ref})$$

with the time-varying reference input $u_{ref} \in \mathbb{R}^2$, the time-varying feedback matrix $K \in \mathbb{R}^{2\times 5}$, and the measured state $\widehat{x} := x + z$ defined using the measurement error $z \in \mathbb{R}^5$. Thus, the ten-dimensional closed-loop system $f(x, u, w)$ is obtained by inserting the control law into the five-dimensional model:

$$\begin{cases} \dot{x} = f(x, u_{ref} + K(x + z - x_{ref}), w) \\ \dot{x}_{ref} = f(x_{ref}, u_{ref}, 0) \end{cases}$$

Table 1: Results of TRAF22 in terms of computation time and verification.

tool	**computation time in [s]**	**Verified?**
Ariadne	N/A	N/A
CORA	150	Yes
DynIbex	357	Yes
JuliaReach	168	Yes
KeYmaera X	27	Yes[1]

[1] Input constraint specification for all possible tracking controllers

3.1.2 Analysis

The set for the measurement error $\mathcal{Z} \subset \mathbb{R}^5$, the input set $\mathcal{U} \subset \mathbb{R}^2$, and the set of disturbances $\mathcal{W} \subset \mathbb{R}^2$ are respectively bounded by

$$\mathcal{Z} = \begin{pmatrix} [-0.0004, 0.0004] \\ [-0.0004, 0.0004] \\ [-0.006, 0.006] \\ [-0.002, 0.002] \\ [-0.002, 0.002] \end{pmatrix} \qquad \mathcal{U} = \begin{pmatrix} [-0.7, 0.7] \\ [-11, 11] \end{pmatrix} \qquad \mathcal{W} = \begin{pmatrix} [-0.02, 0.02] \\ [-0.3, 0.3] \end{pmatrix}.$$

The initial state is uncertain within the set $x_0 \oplus \mathcal{Z} \times x_0$. The inputs u_1, u_2 and the disturbances w_1, w_2 can change arbitrarily over time within their respective sets.

In this case, we analyze the scenario with the identifier *BEL_Putte-4_2_T-1*: The time horizon is determined by the length of the piecewise-constant control values, i.e., the reference trajectory x_{ref}, reference input u_{ref}, and feedback matrix K. All of these are provided by a .csv-file in a format as detailed in [37, Sec. 5].

The following two specifications have to be satisfied:

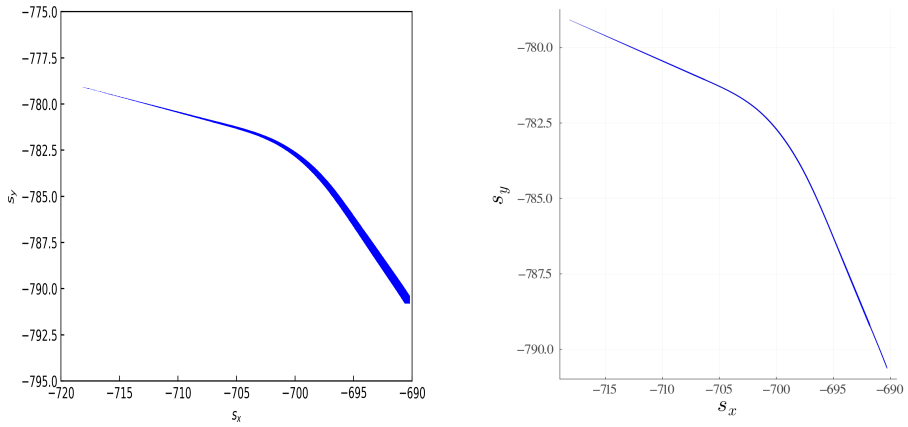
- **Input constraints**: The controller input $u_{fb} \in \mathbb{R}^2$ should be contained within the input set $\mathcal{U}$ at all times. The set of control inputs is computed according to [37, Eq. (5)].
- **Collision avoidance**: The car should not collide with static or dynamic obstacles as well as the road boundaries. Therefore, one requires to compute the car's occupancy set according to [37, Eq. (4)]. After rewriting the occupancy set as a .csv-file using the format in [37, Fig. 4], the collision check is performed fully automatically by calling a provided Python script as detailed in [37, Sec. 5].

3.1.3 Evaluation

There are two metrics to evaluate the performance of each tool. First, we measure the computation time only comprising the time spent during the reachable set computation, exempt the time step in the pre- and post-processing steps. Second, we explicitly tabulate the results of the verification since a collision could occur at any time and therefore might not be captured in the figures below.

3.1.4 Results

The results from this benchmark are shown in Table 1. Some of the tools still do not support the format required by the benchmark.



(a) DynIbex (b) JuliaReach

Figure 1: Reachable set overapproximations for TRAF22.

Settings for Ariadne. Ariadne is currently able to express disturbances within purely continuous dynamics, while the piecewise-constant input requires extension to the hybrid space. We plan on supporting hybrid systems for the next year.

Settings for CORA. We used the conservative linearization approach [12] with a time step size of $\Delta t = 0.005$, resulting in 20 steps per piecewise-constant input. Despite the relatively large system dimension, a zonotope order of 20 was sufficient for a successful verification.

Settings for DynIbex. The Runge-Kutta method selected is at order four (called RK4 in DynIbex). The absolute precision is 10^{-12}. The noise number for affine arithmetic is 200.

Settings for JuliaReach. We use the `TMJets21b` algorithm with $n_Q = 1$, $n_T = 5$, and adaptive absolute tolerance $3 \cdot 10^{-11}$. JuliaReach does not support time-varying disturbances; the disturbances are instead modeled as uncertain but constant state variables $w(0) \in W, \dot{w} = 0$. The reported time consists of computing the reachable states and checking the input constraints.

Settings for KeYmaera X. KeYmaera X models time-varying control references, disturbances, and measurement errors as non-deterministic inputs, and certifies that the tracking controller stays within the input set. In future editions, we plan to obtain monitoring conditions [41] from the formal model to compute occupancy sets.

3.2 Robertson chemical reaction benchmark (ROBE25)

3.2.1 Model

As proposed by Robertson [48], this chemical reaction system models the kinetics of an autocatalytic reaction.

$$\begin{cases} \dot{x} = -\alpha x + \beta yz \\ \dot{y} = \alpha x - \beta yz - \gamma y^2 \\ \dot{z} = \gamma y^2 \end{cases}$$

where x, y and z are the (positive) concentrations of the species, with the assumption that $x+y+z=1$. Here α is a small constant, while β and γ take on large values. In this benchmark we fix $\alpha = 0.4$ and analyze the system under three different pairs of values for β and γ:

1. $\beta = 10^2$, $\gamma = 10^3$
2. $\beta = 10^3$, $\gamma = 10^5$
3. $\beta = 10^3$, $\gamma = 10^7$

The initial condition is always $x(0) = 1$, $y(0) = 0$ and $z(0) = 0$.

3.2.2 Analysis

We are interested in computing the reachable tube until $t = 40$, to see how the integration scheme holds under the stiff behavior. This year we introduced an additional verification target: the width of the sum of the concentrations $s = x + y + z$ at $t = 40s$ must be lower than 10^{-5}.

3.2.3 Evaluation

For each of the three setups, the following three measures are collected:

1. the execution time for evolution;
2. the number of integration steps taken;
3. the width of $x + y + z$ at $t = 40s$ (for reference).

The intent is to enforce the width of s to get to just below 10^{-5} and consequently enable a comparison on execution times and number of integration steps on a common ground.

Additionally, a figure with s (in the $[0.9996, 1.0004]$ range) w.r.t. time overlaid for the three setups is shown to assess convergence speed. Graphical results are shown even if the verification target could not be hit.

3.2.4 Results

All tools were able to get to completion. However, very different results were obtained. In the case of Ariadne and JuliaReach, the width started small and increased monotonically, while for DynIbex and CORA the width started decreasing from a given value. It is also interesting to analyze the number of integration steps taken, which turned out to be sensibly lower for JuliaReach and CORA. While JuliaReach obtained the best width for the stiffest case, this came at the expense of a significantly higher computation time. Perhaps for the next year some verification constraints should be enforced, in order to provide a better baseline for comparison between the tools.

Settings for Ariadne. A GradedTaylorSeriesIntegrator is used, with a maximum error per integration step of 10^{-10}. A maximum step size of 0.004 is imposed in all three setups, though the actual value dynamically identified along evolution for (2) and (3) is sensibly lower.

Settings for CORA. In all cases, we used the approach from [54], which adaptively tunes all algorithm parameters during runtime. To get as close to the desired precision as possible, we manually set the parameter ζ_z from [54, Eq. 49] to $2.5 \times 10^{-}$, 6×10^{-4}, and 0.017 for instances 1-3, respectively.

Table 2: Results of ROBE25 in terms of computation time, number of steps and width of $s = x + y + z$.

	computation time in [s]		
tool	(1)	(2)	(3)
Ariadne	110	483	612
CORA	138	224	321
DynIbex	522	4729	6287
JuliaReach	69	911	3810
KeYmaera X[1]	0.5	0.5	0.5

[1] Single symbolic proof solves all 3 examples

	number of steps		
tool	(1)	(2)	(3)
Ariadne	10000	49849	123675
CORA	16501	22494	25960
DynIbex	13949	114836	147066
JuliaReach	5895	30239	71117
KeYmaera X[1]	326	326	326

[1] Proof steps, symbolic proof solves all 3 examples

	width of $x + y + z$		
tool	(1)	(2)	(3)
Ariadne	2.5e-6	9.4e-6	3.8e-6
CORA	5.8e-6	8.8e-6	8.3e-6
DynIbex	9.6e-6	7.9e-6	7.6e-6
JuliaReach	9.1e-6	3.1e-6	7.6e-11
KeYmaera X[1]	0	0	0

[1] Exact computation without overapproximation

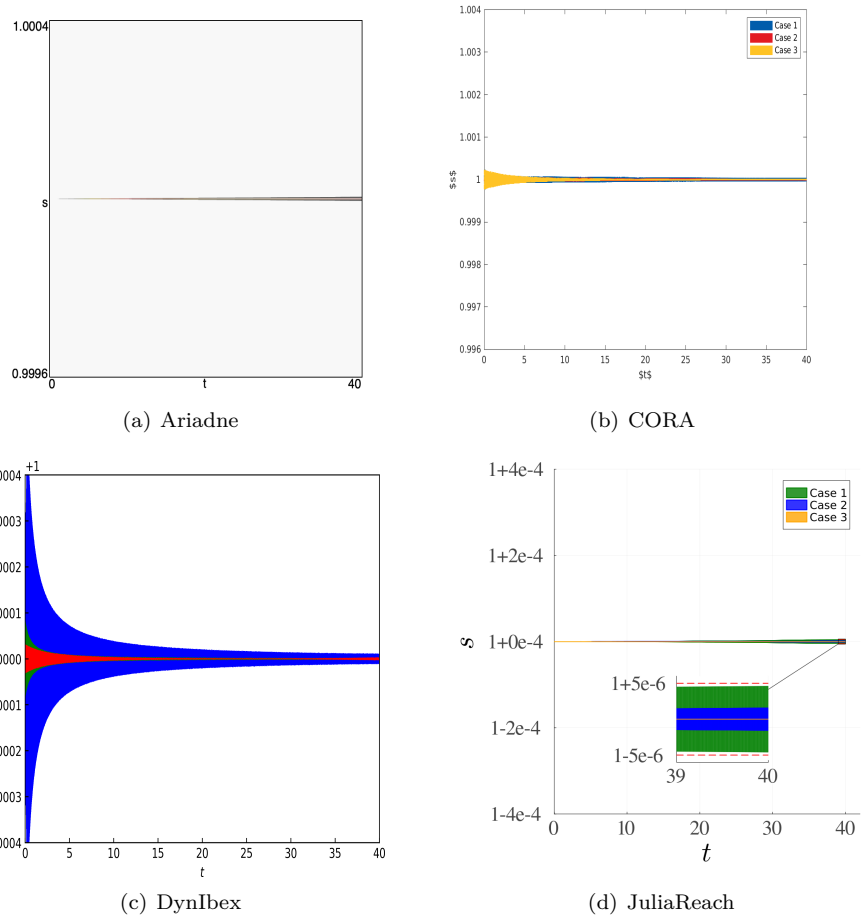
Settings for DynIbex. The Runge-Kutta method selected is implicit Lobatto at fourth order (called LC3 in DynIbex) for the three setups. The absolute precision is, respectively, 1.3e−13, 5e−15 and 3e−14. The other parameters are set by default.

Settings for JuliaReach. In all cases we use the `TMJets21a` algorithm and $n_Q = 1$, and we vary the n_T parameter and the adaptive absolute tolerance as follows: (1) $n_T = 5$ and 10^{-13}, (2) $n_T = 6$ and 10^{-9}, and (3) $n_T = 9$ and 10^{-12}. The maximum number of integration steps is also adjusted, reflecting the results presented in Table 2. For the results displayed in Fig. 2, we evaluate s directly on the Taylor models produced by the integration.

Settings for KeYmaera X. The KeYmaera X proof is fully parametric, without approximation, and shows stability of all possible population sums s for any (even negative) choice of a, b, and g, which includes the specific parametrizations (1) $b = 10^2, g = 10^3$, (2) $b = 10^3, g = 10^5$, and (3) $b = 10^3, g = 10^7$.

```
Problem
  x+y+z=s
  ->
  [{ x' = -a*x + b*y*z,
     y' = a*x - b*y*z - g*y^2,
     z' = g*y^2
   }
  ](x+y+z=s)
End.

Tactic "Scripted proof" unfold; dIClose(1) End.
Tactic "Automated proof" autoClose End.
```



(a) Ariadne (b) CORA

(c) DynIbex (d) JuliaReach

Figure 2: Reachable set overapproximations of $s = x + y + z$ vs time for ROBE25 in the three setups.

3.3 Coupled van der Pol benchmark (CVDP23)

3.3.1 Model

The original van der Pol oscillator was introduced by the Dutch physicist Balthasar van der Pol. For this benchmark we consider two coupled oscillators, as described in [14]. The system can be defined by the following ODE with 5 variables:

$$\begin{cases} \dot{x}_1 = y_1 \\ \dot{y}_1 = \mu(1 - x_1^2)y_1 + b(x_2 - x_1) - x_1 \\ \dot{x}_2 = y_2 \\ \dot{y}_2 = \mu(1 - x_2^2)y_2 - b(x_2 - x_1) - x_2 \\ \dot{b} = 0 \end{cases} \tag{1}$$

with $\mu = 1$. The system has a stable limit cycle that becomes increasingly sharper for higher values of μ.

3.3.2 Analysis

We set the initial condition $x_{1,2}(0) \in [1.25, 1.55]$, $y_{1,2}(0) \in [2.35, 2.45]$ and $b \in [1\,, 3]$. The unsafe set is given by $y_{1,2} \geq 2.75$ in a time horizon of $[0, 7]$.

3.3.3 Evaluation

The computation time required to evolve the system and verify safety is provided. If the system cannot be verified successfully, no value is given.

3.3.4 Results

The computation results of the tools are given in Table 3. While KeYmaera X was not able to participate in this specific benchmark, Ariadne and Cora encountered numerical problems that prevented completion in a reasonable time. Only JuliaReach was able to address the benchmark properly. DynIbex used a partial worst-case analysis to obtain a result in a reasonable time.

Table 3: Results of CVDP23 in terms of computation time.

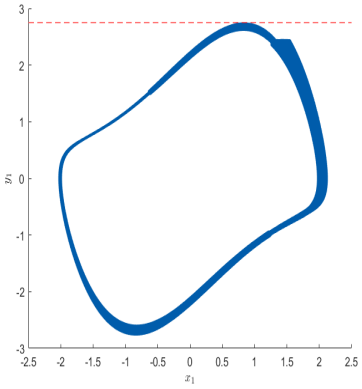
tool	**computation time in [s]**
Ariadne	N/A
CORA	526
DynIbex	990
JuliaReach	1.5
KeYmaera X[1]	35

[1] Simplified $m = 1, b = 1$ and $t \in [0, 0.1]$

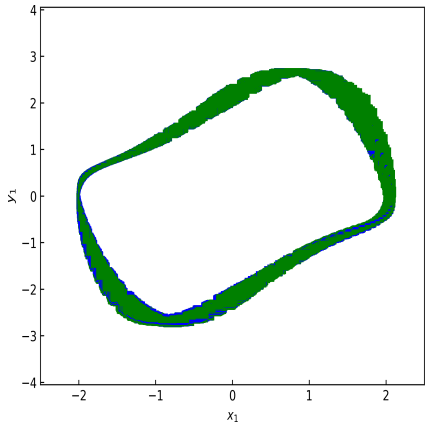
Settings for Ariadne. It was not possible to achieve completion is a reasonable time, due to the very high number of splittings theoretically required to guarantee numerical convergence.

Settings for CORA. Due to the strong nonlinearity induced by the parameter b, it was necessary to split the initial set into 13 smaller subsets. For each run, we used the polynomialization algorithm in [6] with a time step size of 0.005 and a zonotope order of 100. Additionally, we manually introduced two artificial guard sets orthogonal to the flow in order to shrink the reachable set. Otherwise, the abstraction error and thus the computed reachable set would explode in size.

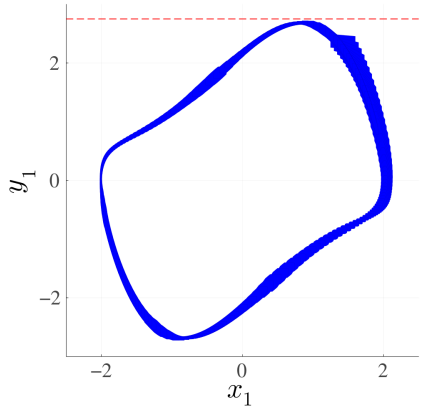
Settings for DynIbex. Maximum zonotope order is set to 80, reachability analysis is carried out with an (absolute and relative) error tolerance of 10^{-6} using an explicit RK4 method of order 4. A formal B-series, based on recent developments [2], is computed with the help of Bseries Julia package. A partial worst case analysis is performed by considering initial value set at extremal value for one dimension w.r.t. others given in intervals. It leads to 5 initial conditions that must be verified. Moreover, a bissection is performed when safety cannot be verified (363 simulations are needed).



(a) CORA



(b) DynIbex



(c) JuliaReach

Figure 3: Reachable set overapproximations for CVDP23.

Settings for JuliaReach. We use the `TMJets21b` algorithm with $n_Q = 1$, $n_T = 4$, and adaptive absolute tolerance 10^{-4}.

Settings for KeYmaera X. The Coupled van der Pol benchmark was formalized for KeYmaera X but not fully verified. Below, we give the formal specification in KeYmaera X format:

```
Definitions Real m, b; End.
ProgramVariables Real x1, x2, y1, y2; End.
Problem
    1 <= b & b <= 3                                                    /* b in  [1,3]  */
  & m = 1
  & 1.25 <= x1 & x1 <= 1.55 & 1.25 <= x2 & x2 <= 1.55 /* x_{1,2}(0) in  [1.25,1.55]  */
  & 2.35 <= y1 & y1 <= 2.45 & 2.35 <= y2 & y2 <= 2.45 /* y_{1,2}(0) in  [2.35,2.45]  */
  & t = 0
  ->
  [{ x1' = y1,
     y1' = m*(1-x1^2)*y1 + b*(x2-x1) - x1,
     x2' = y2,
     y2' = m*(1-x2^2)*y2 - b*(x2-x1) - x2,
     t' = 1 & t <= 7                                                   /* time horizon  [0,7]  */
   }
```

```
    ]!( y1>=2.75 & y2>=2.75)                                    /* not in unsafe set */
End.
```

We simplified the specification to $m = 1$, $b = 1$, rephrased the intervals, and certified a Taylor model to verify safety for a time horizon $[0, 0.1]$. Below, we list the adjusted specification:

```
Problem
( t=0
    & x1 = 0.15*r0()+  0*r1()+   0*r2()+   0*r3() + 1.4   /* x1 in  [1.25,1.55]  */
    & y1 =   0*r0()+   0*r1()+0.05*r2()+  0*r3() + 2.4    /* y1 in  [2.35,2.45]  */
    & x2 =   0*r0()+0.15*r1()+  0*r2()+   0*r3() + 1.4    /* x2 in  [1.25,1.55]  */
    & y2 =   0*r0()+   0*r1()+   0*r2()+0.05*r3() + 2.4)  /* y2 in  [2.35,2.45]  */
    & (-1 <= r0() & r0() <= 1)
    & (-1 <= r1() & r1() <= 1)
    & (-1 <= r2() & r2() <= 1)
    & (-1 <= r3() & r3() <= 1)
  ->
  [{ x1' = y1,
     y1' = (1-x1^2)*y1 - 2*x1 + x2,
     x2' = y2,
     y2' = (1-x2^2)*y2 - 2*x2 + x1,
     t' = 1
     & 0<=t & t<=0+0.1
   }
  ]!( y1>=2.75 & y2>=2.75)
End.
```

The tactic to perform Taylor model certification is an early preview and needs performance improvements; it performs $54 \cdot 10^6$ proof steps to certify the simplified model with $m = 1, b = 1, t \in [0, 0.1]$.

3.4 Laub-Loomis benchmark (LALO20)

3.4.1 Model

The Laub-Loomis model is presented in [38] for studying a class of enzymatic activities. The dynamics can be defined by the following ODE with 7 variables.

$$\begin{cases} \dot{x}_1 = 1.4x_3 - 0.9x_1 \\ \dot{x}_2 = 2.5x_5 - 1.5x_2 \\ \dot{x}_3 = 0.6x_7 - 0.8x_2x_3 \\ \dot{x}_4 = 2 - 1.3x_3x_4 \\ \dot{x}_5 = 0.7x_1 - x_4x_5 \\ \dot{x}_6 = 0.3x_1 - 3.1x_6 \\ \dot{x}_7 = 1.8x_6 - 1.5x_2x_7 \end{cases}$$

The system is asymptotically stable, with the equilibrium point approximately [-0.87, 0.37, -0.56, -2.75, 0.22, -0.08, -0.27].

3.4.2 Analysis

The specification for the analysis is kept the same as last year, in order to better quantify any improvements to the participating tools.

The initial sets are defined according to the ones used in [52]. They are boxes centered at $x_1(0) = 1.2$, $x_2(0) = 1.05$, $x_3(0) = 1.5$, $x_4(0) = 2.4$, $x_5(0) = 1$, $x_6(0) = 0.1$, $x_7(0) = 0.45$. The range of the box in the ith dimension is defined by the interval $[x_i(0) - W, x_i(0) + W]$. The width W of the initial set is vital to the difficulty of the reachability analysis job. The larger the initial set the harder the reachability analysis.

We consider $W = 0.01$, $W = 0.05$, and $W = 0.1$, leading to three instances called $W001$, $W005$ and $W01$ respectively.

For $W001$ and $W005$ we consider the unsafe region defined by $x_4 \geq 4.5$, while for $W01$, the unsafe set is defined by $x_4 \geq 5$. The time horizon for all cases is $[0, 20]$.

3.4.3 Evaluation

The final widths of x_4 along with the computation times are provided for all three cases. A figure is provided in the (t, x_4) axes, with $t \in [0, 20]$, $x_4 \in [1.5, 5]$, where the three plots are overlaid.

3.4.4 Results

The computation results of the tools are given in Table 4. The results are essentially identical to last year's.

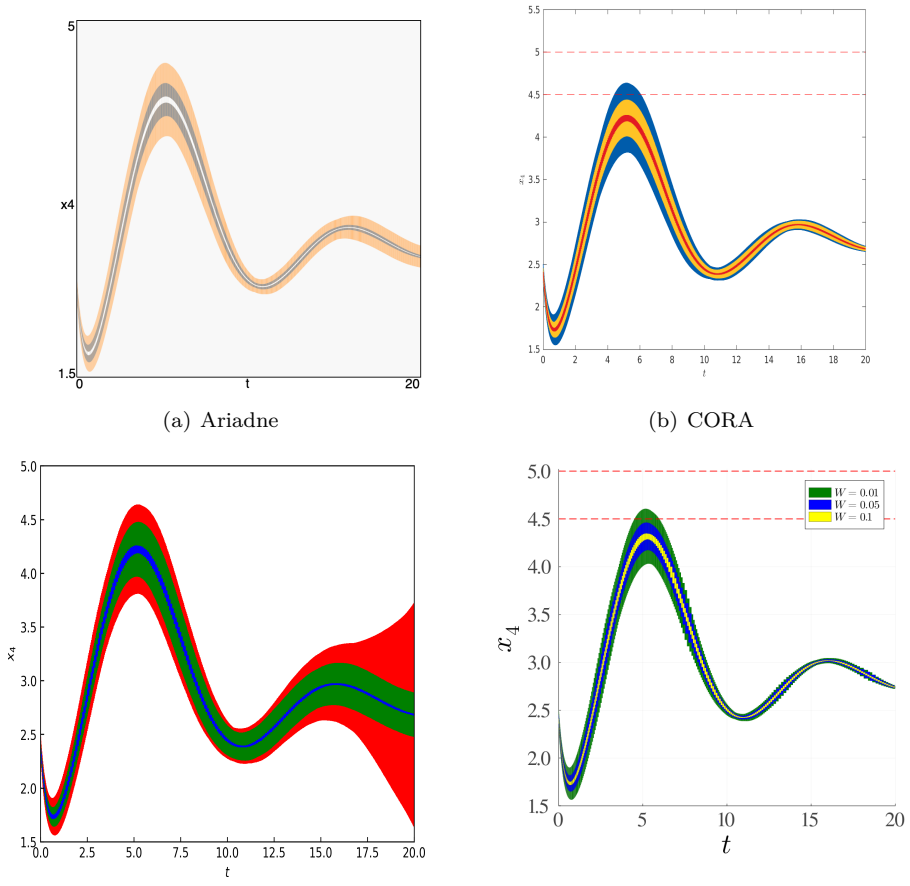
Table 4: Results of LALO20 in terms of computation time and width of final enclosure.

	computation time in [s]		
tool	$W001$	$W005$	$W01$
Ariadne	35	113	305
CORA	9.1	16	242
DynIbex	12	30	1942
JuliaReach	5.5	4.1	5.1
KeYmaera X	N/A	N/A	229[1]

[1] Simplified model with $t \in [0, 0.1]$

	width of x_4 in final enclosure		
tool	$W001$	$W005$	$W01$
Ariadne	0.004	0.029	0.21
CORA	0.004	0.042	0.060
DynIbex	0.01	0.40	2.07
JuliaReach	0.003	0.019	0.031
KeYmaera X	N/A	N/A	N/A

Settings for Ariadne. The maximum step size used is 0.2, with a TaylorPicardIntegrator with a maximum error of 10^{-7} enforced for each step and a maximum spacial error of 10^{-4}.



(a) Ariadne (b) CORA

(c) DynIbex (d) JuliaReach

Figure 4: Reachable set overapproximations for LALO20 (overlayed plots for $W001$, $W005$, $W01$). $t \in [0, 20]$, $x_4 \in [1.5, 5]$.

Settings for CORA. For the smaller initial sets $W001$ and $W005$, we applied an adaptively-tuned linearization algorithm [54], whereas the larger initial set $W01$ required a polynomialization algorithm, where we again used the adaptively-tuned version from [54].

Settings for DynIbex. An explicit RK4 method of order 4 is used. A formal B-series, based on [2], is computed with the help of Bseries Julia package. For $W001$ the maximum zonotope order is set to 50 and the reachability analysis is carried out with an (absolute and relative) error tolerance of 10^{-6}. For $W005$ the maximum zonotope order is set to 80 and the reachability analysis is carried out with an (absolute and relative) error tolerance of 10^{-7}. For $W001$ and $W005$ no splitting of the initial conditions is performed. For $W01$, the initial set is split 64 times. With parallelization, the computation time is reduced for the three instances.

Settings for JuliaReach. In all cases, we use the `TMJets21b` algorithm with $n_Q = 1$, and we vary the n_T parameter and the adaptive absolute tolerance as follows. $W001$: $n_T = 7$ and $5 \cdot 10^{-11}$; $W005$: $n_T = 3$ and $5 \cdot 10^{-4}$; $W01$: $n_T = 3$ and $3 \cdot 10^{-4}$.

Settings for KeYmaera X. The Laub-Loomis benchmark was formalized for KeYmaera X but not fully verified. Below, we give the formal specification in KeYmaera X format:

```
Definitions
  Real W = 0.1;
  Bool box(Real x, Real c, Real w) <-> c-w <= x & x <= c+w;
End.

ProgramVariables
  Real x1, x2, x3, x4, x5, x6, x7;  /* state space */
  Real t;                           /* time */
End.

Problem
    box(x1, 1.2, W)                 /* initial sets */
  & box(x2, 1.05, W)
  & box(x3, 1.5, W)
  & box(x4, 2.4, W)
  & box(x5, 1, W)
  & box(x6, 0.1, W)
  & box(x7, 0.45, W)
  & t=0
  ->
  [{ x1' = 1.4*x3 - 0.9*x1,
     x2' = 2.5*x5 - 1.5*x2,
     x3' = 0.6*x7 - 0.8*x2*x3,
     x4' = 2 - 1.3*x3*x4,
     x5' = 0.7*x1 - x4*x5,
     x6' = 0.3*x1 - 3.1*x6,
     x7' = 1.8*x6 - 1.5*x2*x7,
     t' = 1 & t<=20                 /* time horizon [0,20] */
   }
  ]!( x4>=5)                        /* not in unsafe set */
End.
```

We focused on $W = 0.1$ rephrased the intervals and certified a Taylor model to verify safety for a time horizon $[0, 0.1]$. Below, we list the adjusted specification:

```
Problem
  (t=0
   & x1 = W*r0() + 0*r1() + 0*r2() + 0*r3() + 0*r4() + 0*r5() + 0*r6() + 1.2
   & x2 = 0*r0() + W*r1() + 0*r2() + 0*r3() + 0*r4() + 0*r5() + 0*r6() + 1.05
   & x3 = 0*r0() + 0*r1() + W*r2() + 0*r3() + 0*r4() + 0*r5() + 0*r6() + 1.5
   & x4 = 0*r0() + 0*r1() + 0*r2() + W*r3() + 0*r4() + 0*r5() + 0*r6() + 2.4
   & x5 = 0*r0() + 0*r1() + 0*r2() + 0*r3() + W*r4() + 0*r5() + 0*r6() + 1.0
   & x6 = 0*r0() + 0*r1() + 0*r2() + 0*r3() + 0*r4() + W*r5() + 0*r6() + 0.1
   & x7 = 0*r0() + 0*r1() + 0*r2() + 0*r3() + 0*r4() + 0*r5() + W*r6() + 0.45)
   & (-1 <= r0() & r0() <= 1)
   & (-1 <= r1() & r1() <= 1)
   & (-1 <= r2() & r2() <= 1)
   & (-1 <= r3() & r3() <= 1)
   & (-1 <= r4() & r4() <= 1)
   & (-1 <= r5() & r5() <= 1)
   & (-1 <= r6() & r6() <= 1)
  ->
  [{ x1' = 1.4*x3 - 0.9*x1,
     x2' = 2.5*x5 - 1.5*x2,
     x3' = 0.6*x7 - 0.8*x2*x3,
     x4' = 2 - 1.3*x3*x4,
     x5' = 0.7*x1 - x4*x5,
     x6' = 0.3*x1 - 3.1*x6,
     x7' = 1.8*x6 - 1.5*x2*x7,
     t' = 1
     & 0<=t & t<=0+0.1
   }
```

```
]!( x4>=5)
End.
```

The tactic to perform Taylor model certification is an early preview and needs performance improvements; it performs $23 \cdot 10^6$ proof steps to certify the simplified model with $t \in [0, 0.1]$.

3.5 Lotka–Volterra with tangential crossings benchmark (LOVO25)

3.5.1 Model

The benchmark described below refers to the Lotka-Volterra equations, or predator-prey equations, which are well-known in the literature.

The system is defined as follows:

$$\begin{cases} \dot{x} = 3x - 3xy \\ \dot{y} = xy - y \end{cases} \tag{2}$$

which produces cyclic trajectories around the equilibrium point $(1, 1)$ dependent on the initial state.

We are interested to see how this nonlinear dynamics plays with a nonlinear guard, whose boundary is:

$$\sqrt{(x-1)^2 + (y-1)^2} = 0.161 \tag{3}$$

which is a circle of radius 0.161 around the equilibrium.

By choosing an initial state $I = (1.3, 1.0)$ the cycle has a period of approximately 3.64 time units. The trajectory of the Lotka–Volterra system trajectory is close to tangent to the guard circle in the top half, while it crosses the circle on the bottom half. Hence, enlarging the width of the initial set would put the trajectory partially within the guard in the top half.

The corresponding hybrid automaton is used to model the system:

- Continuous variables: x, y;
- Locations: *outside* and *inside*;
- Dynamics: those from Eq. 2 for x, y in both locations;
- Guards:
$$\begin{cases} (x - Q_x)^2 + (y - Q_y)^2 \leq R^2 \text{ from } \textit{outside} \text{ to } \textit{inside} \\ (x - Q_x)^2 + (y - Q_y)^2 \geq R^2 \text{ from } \textit{inside} \text{ to } \textit{outside} \end{cases} \tag{4}$$
- Invariants: the complement of the corresponding guards (i.e., transitions are urgent);
- Resets: none, i.e., the identity for both transitions.

3.5.2 Analysis

We want to start the system from $I = (1.3 \pm \epsilon, 1.0)$, with $\epsilon = 0.012$, and evolve it for $T = 3.64$ time units. Since the original system was close to tangency, by enlarging the initial set we expect to produce different sequences of discrete events due to the distinction between crossing and not crossing, and possibly by distinguishing the crossing sets based on the different crossing times. We must remark that, for reachability analysis purposes, it is important to carry the trace of discrete events along with the current evolution time.

The following four properties must be verified:

Table 5: Results of LOVO25 in terms of computation time and area.

tool	**computation time in [s]**	**area**
Ariadne	96	2.5e-4
CORA	8.8	8.9e-3
DynIbex	23	6.4e-4
JuliaReach	1.2	9e-5
KeYmaera X	(5.8)[1]	0

[1] Duration of proving invariance (not checking crossing)

- The area $x \times y$ of the box hull enclosing all the final sets must be lower than 10^{-2};
- At least one final set must have crossed two guards by entering and exiting the reference circle once;
- At least one final set must have crossed four guards by entering and exiting the reference circle twice;
- While a larger *even* number of crossings is allowed due to Zeno behavior during tangent crossing, no odd numbers are possible.

3.5.3 Evaluation

In terms of metrics, it is required to supply the following:

1. The execution time for computing the reachable set and checking the properties;
2. The area $x \times y$ of the box hull enclosing all the final sets.

In addition, a figure showing the reachable set along with the circular guard shall be provided. The axes are $[0.6, 1.4] \times [0.6, 1.4]$.

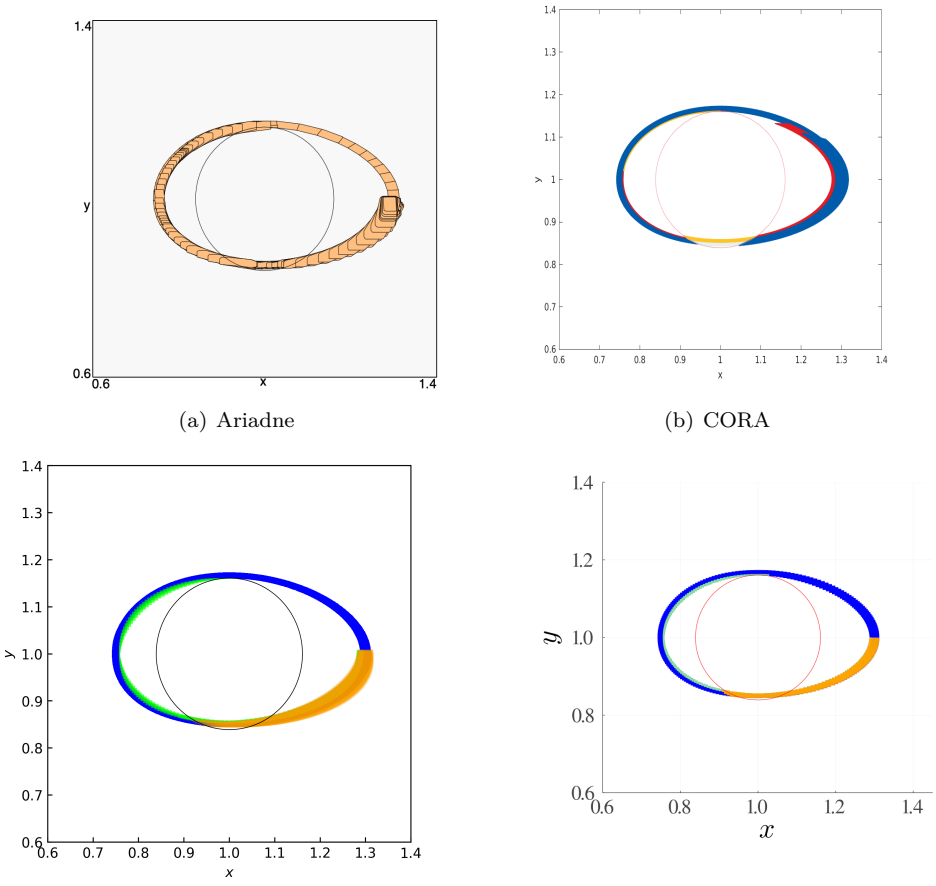
3.5.4 Results

All tools were able to handle the benchmark with results equivalent to last year. Table 5 gives the timing/quality results, while Fig. 5 shows the graphical output.

Settings for Ariadne. A GradedTaylorSeriesIntegrator is used with a maximum spacial error of 10^{-4} and maximum step error of 10^{-7}. The maximum step size is 0.2.

Settings for CORA. We use the approach in [36] to compute the intersections with the nonlinear guard set. For continuous reachability we apply the conservative linearization approach [12] with time step size of 0.02 and a zonotope order of 20 for all modes.

Settings for DynIbex. The library DynIbex does not support hybrid systems natively. However, based on constraint programming, event detection can be implemented and hybrid systems can be simulated. Reachability analysis is carried out with an error tolerance of 10^{-14} using an explicit Runge-Kutta method of order 4 (RK4 method). No splitting of the initial state has been performed.



(a) Ariadne (b) CORA

(c) DynIbex (d) JuliaReach

Figure 5: Reachable set overapproximation for LOVO25, with $x, y \in [0.6, 1.4]$, where the circular guard is shown.

Settings for JuliaReach. We use the `TMJets21b` algorithm with $n_T = 3$, $n_Q = 1$, and adaptive absolute tolerance 10^{-4}. We also split the initial set into 5 intervals in each dimension. The crossings to the nonlinear guard are handled by checking the reach sets that do not lie strictly outside the circle.

Settings for KeYmaera X. The KeYmaera X proof focuses on infinite-horizon population stability for any positive starting choice of populations $x > 0$ and $y > 0$, which includes the specific starting populations $x = 1.3 \pm \epsilon$ and $y = 1$. The population orbit is stable around $(\frac{\alpha}{\beta}, \frac{\gamma}{\delta})$ at population $e^{-\delta x - \beta y} x^\gamma y^\alpha$ for $\alpha = \beta = 3$ and $\delta = \gamma = 1$.

```
1  Definitions Real K(Real x, Real y) = exp(−d∗x−b∗y) ∗ x^g ∗ y^a; End.
2  Problem
3    a=3 & b=3 & d=1 & g=1 & x>0 & y>0 & K_0 = K(x,y)
4    −>
5    [{ x' = a∗x − b∗x∗y,
6       y' = d∗x∗y − g∗y
```

```
    }
  ]K(x,y) = K_0
End.

Tactic "Scripted proof"
useSolver("Mathematica");
unfold;
dIRule(1); <(
  "dI Init": equalCommute(1); id,
  "dI Step":
    chaseAt(1);
    QE using "(exp(-1*x-3*y)*(-1*(3*x-3*x*y)-3*(1*x*y-1*y))*x^1+exp(-1*x-3*y)*(1*x^(1-1)*(3*x
        ↪ -3*x*y)))*y^3+exp(-1*x-3*y)*x^1*(3*y^(3-1)*(1*x*y-1*y))=0"
)
End.
Tactic "Automated proof" autoClose End.
```

The formalization in the repeatability package also includes a symbolic characterization of the existence of crossing in and out of the nonlinear guard: this purely real arithmetic proof obligation is not yet tractable by the arithmetic backend verification procedures used in KeYmaera X. In future editions, we plan to additionally characterize the number of transitions symbolically.

3.6 Space rendezvous benchmark (SPRE22)

3.6.1 Model

Space rendezvous is a perfect use case for formal verification of hybrid systems with nonlinear dynamics since mission failure can cost lives and is extremely expensive. This benchmark is taken from [27]. A version of this benchmark with linearized dynamics is verified in the ARCH-COMP category *Continuous and Hybrid Systems with Linear Continuous Dynamics*. The nonlinear dynamic equations describe the two-dimensional, planar motion of the spacecraft on an orbital plane towards a space station:

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{v_x} = n^2 x + 2nv_y + \dfrac{\mu}{r^2} - \dfrac{\mu}{r_c^3}(r + x) + \dfrac{u_x}{m_c} \\ \dot{v_y} = n^2 y - 2nv_x - \dfrac{\mu}{r_c^3} y + \dfrac{u_y}{m_c} \end{cases}$$

The model consists of position (relative to the target) x, y [m], time t [min], as well as horizontal and vertical velocity v_x, v_y [m / min]. Nonlinearity comes from the variable term $r_c = \sqrt{(r+x)^2 + y^2}$. The parameters are $\mu = 3.986 \times 10^{14} \times 60^2$ [m^3 / min^2], $r = 42164 \times 10^3$ [m], $m_c = 500$ [kg] and $n = \sqrt{\frac{\mu}{r^3}}$.

The hybrid nature of this benchmark originates from a switched controller. In particular, the modes are *approaching* ($x \in [-1000, -100]$ [m]), *rendezvous attempt* ($x \geq -100$ [m]), and *aborting*. A transition to mode *aborting* occurs nondeterministically at $t \in [120, 150]$ [*min*]. The linear feedback controllers for the different modes are defined as $\binom{u_x}{u_y} = K_1 \underline{x}$ for mode *approaching*, and $\binom{u_x}{u_y} = K_2 \underline{x}$ for mode *rendezvous attempt*, where $\underline{x} = \begin{pmatrix} x & y & v_x & v_y \end{pmatrix}^T$ is the vector of system states. The feedback matrices K_i were determined with an LQR-approach applied to the linearized system dynamics, which resulted in the following numerical values:

$$K_1 = \begin{pmatrix} -28.8287 & 0.1005 & -1449.9754 & 0.0046 \\ -0.087 & -33.2562 & 0.00462 & -1451.5013 \end{pmatrix}$$

$$K_2 = \begin{pmatrix} -288.0288 & 0.1312 & -9614.9898 & 0 \\ -0.1312 & -288 & 0 & -9614.9883 \end{pmatrix}$$

In the mode *aborting*, the system is uncontrolled $\binom{u_x}{u_y} = \binom{0}{0}$.

3.6.2 Analysis

The spacecraft starts from the initial set $x \in [-925, -875]$ [m], $y \in [-425, -375]$ [m], $v_x \in [0, 5]$ [m/min] and $v_y \in [0, 5]$ [m/min]. For the considered time horizon of $t \in [0, 200]$ [min], the following specifications have to be satisfied:

- **Line-of-sight:** In mode *rendezvous attempt*, the spacecraft has to stay inside line-of-sight cone $\mathcal{L} = \{\binom{x}{y} \mid (x \geq -100) \wedge (y \geq x \tan(20°)) \wedge (-y \geq x \tan(20°))\}$.
- **Collision avoidance:** In mode *aborting*, the spacecraft has to avoid a collision with the target, which is modeled as a box $\mathcal{B}$ with 2m edge length and the center placed at the origin.
- **Velocity constraint:** In mode *rendezvous attempt*, the absolute velocity has to stay below 3.3 [m/min]: $\sqrt{v_x^2 + v_y^2} \leq 3.3$ [m/min].

Remark on velocity constraint In the original benchmark [27], the constraint on the velocity was set to 0.05 m/s, but it can be shown (by a counterexample) that this constraint cannot be satisfied. We therefore use the relaxed constraint 0.055 [m/s] = 3.3 [m/min].

3.6.3 Evaluation

The computation time for evolution and verification is provided. A figure is shown in the (x, y) axes, with $x \in [-1000, 200]$ and $y \in [-450, 0]$.

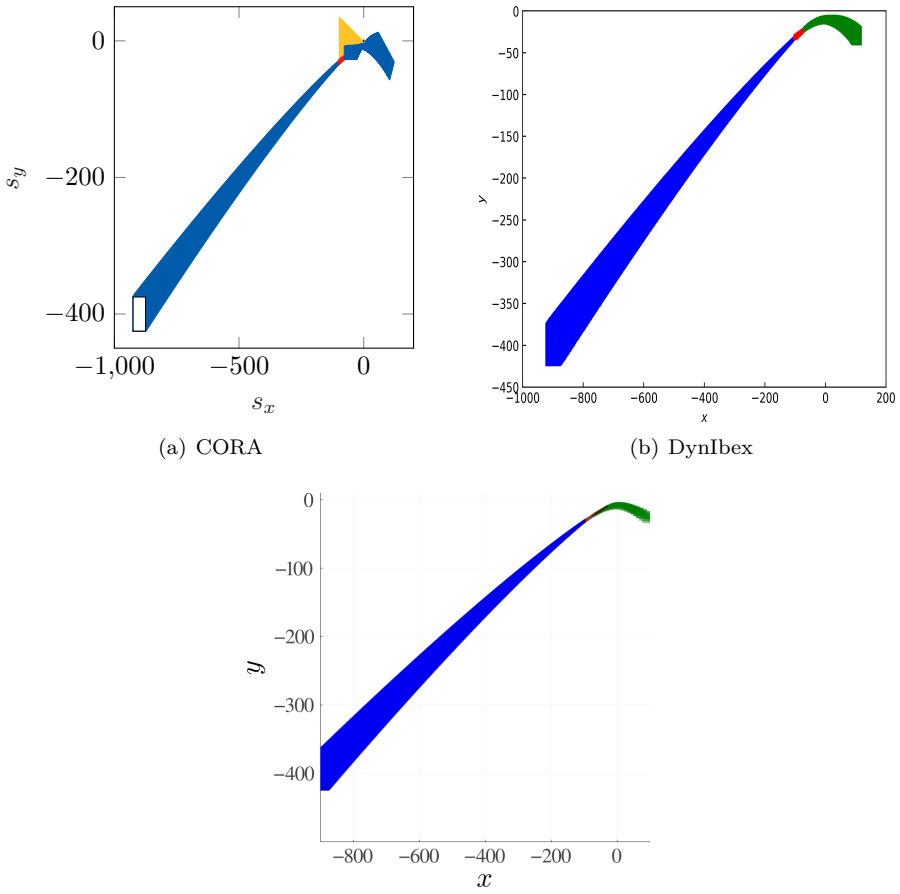
3.6.4 Results

The results of the reachability computation for the space rendezvous model are given in Figure 6 and Table 6, with the tool settings below. The introduction of a permissive guard prevented completion for Ariadne: too many trajectories were generated and the absence of a recombination strategy proved an issue. Therefore, this benchmark requires proper support of crossings in the presence of large sets, even if the crossing region is very simple from a geometrical viewpoint. KeYmaera X formalized but did not prove the problem yet.

Settings for Ariadne. Ariadne was not able to complete evolution, due to the extremely large number of trajectories produced from the nondeterministic guard: this is caused by the lack of a recombination strategy. The maximum step size used was 1.0, essentially meaning that we allowed the step size to vary widely along evolution: this choice turned out to be preferable

Table 6: Results of SPRE22 in terms of computation time.

tool	**computation time in [s]**
Ariadne	–
CORA	35
DynIbex	147
JuliaReach	110
KeYmaera X	N/A



(a) CORA

(b) DynIbex

(c) JuliaReach

Figure 6: Reachable set of the spacecraft position in the x-y-plane for SPRE22.

in terms of execution time. The maximum temporal order was 4 and the maximum spacial error enforced for each step equal is 10^{-3}. A splitting strategy for the initial set was used; the strategy compare the radius of the set with a reference value of 12.0, in order to split the first two dimensions once and yield a total of 4 initial subsets.

Settings for CORA. CORA was run with a time step size of $\Delta t = 0.2$ [min] for the modes *approaching* and *aborting*, and with a time step size of $\Delta t = 0.05$ [min] for mode *rendezvous attempt*. The intersections with the guard sets are calculated with constrained zonotopes [49], and the intersection is then enclosed with a zonotope bundle [11]. We applied *principal component analysis* to find suitable orthogonal directions for the enclosure.

Settings for DynIbex. The library DynIbex does not support hybrid systems natively. However, based on constraint programming, event detection can be implemented and hybrid systems can be simulated. Maximum zonotope order is set to 10, reachability analysis is carried out with an error tolerance of 10^{-6} using an explicit Runge-Kutta method of order 3 (Kutta's method). No splitting of the initial state has been performed.

Settings for JuliaReach. The transition to the aborting mode is handled by clustering and Cartesian decomposition [22] with zonotope enclosures in low dimensions, (x, y) and (v_x, v_y). The continuous-time algorithms used in the modes (*approaching*, *rendezvous attempt*, and *aborting*) are `TMJets21a` (first two modes) and `TMJets21b` (third mode) with $n_T = 3, 5, 7$ and adaptive absolute tolerance $4 \cdot 10^{-4}, 5 \cdot 10^{-6}, 10^{-10}$, respectively, and $n_Q = 1$ in all cases.

Settings for KeYmaera X. The example was formalized for KeYmaera X but not yet proved. The full model is included in the repeatability evaluation package.

3.7 Transient Stability Analysis of Power Systems (TSPS25)

Power systems are a vital part of civil infrastructure, making them a perfect use case for formal verification. We consider the transient stability analysis benchmark introduced in [7]. The benchmark investigates whether a power system is able to reach an acceptable operating condition in which the generators are synchronized after a contingency.

3.7.1 Model

We consider the model of the IEEE 14 bus power system, which is a popular example in power system analysis [45]. We obtain a nonlinear differential algebraic system of the form

$$\dot{x} = f(x(t), y(t), u(t))$$
$$0 = g(x(t), y(t), u(t))$$

representing the power system using the automatic benchmark generation approach from [13]. We provide the differential and algebraic equations as MATLAB scripts for ease of use. The differential dynamics $f(x(t), y(t), u(t))$ are given as

$$
\begin{cases}
\dot{x}_1 = x_6 - 120\pi, \\
\dot{x}_2 = x_7 - 120\pi, \\
\dot{x}_3 = x_8 - 120\pi, \\
\dot{x}_4 = x_9 - 120\pi, \\
\dot{x}_5 = x_{10} - 120\pi, \\
\dot{x}_6 = 15x_{11}\pi - \dfrac{3\pi(x_6 - 120\pi)}{5} - \dfrac{159y_1\pi \sin x_1}{2}, \\
\dot{x}_7 = 15x_{12}\pi - \dfrac{3\pi(x_7 - 120\pi)}{5} - \dfrac{627y_2\pi \sin(x_2 - y_{15})}{8}, \\
\dot{x}_8 = 15x_{13}\pi - \dfrac{3\pi(x_8 - 120\pi)}{5} - \dfrac{303y_3\pi \sin(x_3 - y_{16})}{4}, \\
\dot{x}_9 = 15x_{14}\pi - \dfrac{3\pi(x_9 - 120\pi)}{5} - \dfrac{321y_4\pi \sin(x_4 - y_{17})}{4}, \\
\dot{x}_{10} = 15x_{15}\pi - \dfrac{3\pi(x_{10} - 120\pi)}{5} - \dfrac{327y_5\pi \sin(x_5 - y_{18})}{4}, \\
\dot{x}_{11} = u_1 - p_1 x_6 - x_{11} + p_2, \\
\dot{x}_{12} = u_2 - p_1 x_7 - x_{12} + p_2, \\
\dot{x}_{13} = u_3 - p_1 x_8 - x_{13} + p_2, \\
\dot{x}_{14} = u_4 - p_1 x_9 - x_{14} + p_2, \\
\dot{x}_{15} = u_5 - p_1 x_{10} - x_{15} + p_2.
\end{cases}
$$

with $p_1 = 0.0531$ and $p_2 = 20$, and the algebraic equations are

$$
\left\{
\begin{array}{rcl}
g_1 &=& 17.8\cos(y_{15}+1.89) - 1.0u_6 - 17.8\sin(y_{15}+1.89) + 4.62y_7\cos(y_{20}+1.81) \\
&& -4.62y_7\sin(y_{20}+1.81) - 5.3y_1\cos(x_1) - 5.3y_1\sin(x_1) + 34.2 \\
g_2 &=& 17.8\cos(y_{15}-1.89) - 1.0u_7 + 5.19\cos(y_{16}-y_{15}+1.8) - 5.22y_2\sin(x_2-y_{15}) \\
&& +5.71y_7\cos(y_{20}-y_{15}+1.89) + 5.63y_6\cos(y_{19}-y_{15}+1.89) + 10.6 \\
g_3 &=& 5.19\cos(y_{15}-y_{16}+1.8) - 1.0u_8 - 5.05y_3\sin(x_3-y_{16}) \\
&& +5.5y_6\cos(y_{19}-y_{16}+1.94) + 4.13 \\
g_4 &=& 4.85y_{11}\cos(y_{24}-y_{17}+2.02) - 5.35y_4\sin(x_4-y_{17}) - 1.0u_9 \\
&& +3.77y_{12}\cos(y_{25}-y_{17}+2.02) + 7.32y_{13}\cos(y_{26}-y_{17}+2.04) \\
&& +4.56y_7\cos(y_{20}-y_{17}+1.57) + 7.65 \\
g_5 &=& 6.19y_8\cos(y_{21}-y_{18}+1.57) - 5.45y_5\sin(x_5-y_{18}) - 1.0u_{10} \\
g_6 &=& 5.63y_6\cos(y_{15}-y_{19}+1.89) - 1.0u_{11} + 5.5y_6\cos(y_{16}-y_{19}+1.94) + 10.5y_6^2 \\
&& +4.89y_6y_8\cos(y_{21}-y_{19}+1.57) + 1.86y_6y_9\cos(y_{22}-y_{19}+1.57) \\
&& +22.6y_6y_7\cos(y_{20}-y_{19}+1.88) + 0.478 \\
g_7 &=& 4.62y_7\cos(y_{20}-1.81) - 1.0u_{12} + 4.56y_7\cos(y_{17}-y_{20}+1.57) \\
&& +5.71y_7\cos(y_{15}-y_{20}+1.89) + 9.57y_7^2 + 22.6y_6y_7\cos(y_{19}-y_{20}+1.88) + 0.076 \\
g_8 &=& 6.19y_8\cos(y_{18}-y_{21}+1.57) - 1.0u_{13} + 4.89y_6y_8\cos(y_{19}-y_{21}+1.57) \\
&& +9.09y_8y_9\cos(y_{22}-y_{21}+1.57) \\
g_9 &=& 5.33y_9^2 - 1.0u_{14} + 1.86y_6y_9\cos(y_{19}-y_{22}+1.57) + 9.09y_8y_9\cos(y_{21}-y_{22}+1.57) \\
&& +11.1y_9y_{10}\cos(y_{23}-y_{22}+1.93) + 3.35y_9y_{14}\cos(y_{27}-y_{22}+2.01) + 0.295 \\
g_{10} &=& 5.78y_{10}^2 - 1.0u_{15} + 11.1y_9y_{10}\cos(y_{22}-y_{23}+1.93) \\
&& +4.79y_{10}y_{11}\cos(y_{24}-y_{23}+1.97) + 0.09 \\
g_{11} &=& 4.85y_{11}\cos(y_{17}-y_{24}+2.02) - 1.0u_{16} + 3.84y_{11}^2 + 4.79y_{10}y_{11}\cos(y_{23}-y_{24}+1.97) \\
&& +0.035 \\
g_{12} &=& 3.77y_{12}\cos(y_{17}-y_{25}+2.02) - 1.0u_{17} + 4.01y_{12}^2 + 3.36y_{12}y_{13}\cos(y_{26}-y_{25}+2.41) \\
&& +0.061 \\
g_{13} &=& 7.32y_{13}\cos(y_{17}-y_{26}+2.04) - 1.0u_{18} + 6.72y_{13}^2 + 2.58y_{13}y_{14}\cos(y_{27}-y_{26}+2.03) \\
&& +3.36y_{12}y_{13}\cos(y_{25}-y_{26}+2.41) + 0.135 \\
g_{14} &=& 2.56y_{14}^2 - 1.0u_{19} + 3.35y_9y_{14}\cos(y_{22}-y_{27}+2.01) \\
&& +2.58y_{13}y_{14}\cos(y_{26}-y_{27}+2.03) + 0.149 \\
g_{15} &=& 17.8\sin(y_{15}-1.89) - 5.19\sin(y_{16}-y_{15}+1.8) - 5.22y_2\cos(x_2-y_{15}) \\
&& -5.71y_7\sin(y_{20}-y_{15}+1.89) - 5.63y_6\sin(y_{19}-y_{15}+1.89) + 38.6 \\
g_{16} &=& 15.3 - 5.05y_3\cos(x_3-y_{16}) - 5.5y_6\sin(y_{19}-y_{16}+1.94) - 5.19\sin(y_{15}-y_{16}+1.8) \\
g_{17} &=& 25.7 - 4.85y_{11}\sin(y_{24}-y_{17}+2.02) - 3.77y_{12}\sin(y_{25}-y_{17}+2.02) \\
&& -7.32y_{13}\sin(y_{26}-y_{17}+2.04) - 4.56y_7\sin(y_{20}-y_{17}+1.57) - 5.35y_4\cos(x_4-y_{17}) \\
g_{18} &=& 12.7 - 6.19y_8\sin(y_{21}-y_{18}+1.57) - 5.45y_5\cos(x_5-y_{18}) \\
g_{19} &=& 38.7y_6^2 - 5.5y_6\sin(y_{16}-y_{19}+1.94) - 5.63y_6\sin(y_{15}-y_{19}+1.89) \\
&& -4.89y_6y_8\sin(y_{21}-y_{19}+1.57) - 1.86y_6y_9\sin(y_{22}-y_{19}+1.57) \\
&& -22.6y_6y_7\sin(y_{20}-y_{19}+1.88) - 0.039 \\
g_{20} &=& 4.62y_7\sin(y_{20}-1.81) - 4.56y_7\sin(y_{17}-y_{20}+1.57) - 5.71y_7\sin(y_{15}-y_{20}+1.89) \\
&& +35.5y_7^2 - 22.6y_6y_7\sin(y_{19}-y_{20}+1.88) + 0.016 \\
g_{21} &=& 19.5y_8^2 - 6.19y_8\sin(y_{18}-y_{21}+1.57) - 4.89y_6y_8\sin(y_{19}-y_{21}+1.57) \\
&& -9.09y_8y_9\sin(y_{22}-y_{21}+1.57)
\end{array}
\right.
$$

$$\begin{cases} g_{22} &= 24.1y_9^2 - 1.86y_6y_9\sin(y_{19}-y_{22}+1.57) - 9.09y_8y_9\sin(y_{21}-y_{22}+1.57) \\ & \quad -11.1y_9y_{10}\sin(y_{23}-y_{22}+1.93) - 3.35y_9y_{14}\sin(y_{27}-y_{22}+2.01) + 0.166 \\ g_{23} &= 14.8y_{10}^2 - 11.1y_9y_{10}\sin(y_{22}-y_{23}+1.93) - 4.79y_{10}y_{11}\sin(y_{24}-y_{23}+1.97) + 0.058 \\ g_{24} &= 8.5y_{11}^2 - 4.85y_{11}\sin(y_{17}-y_{24}+2.02) - 4.79y_{10}y_{11}\sin(y_{23}-y_{24}+1.97) + 0.018 \\ g_{25} &= 5.43y_{12}^2 - 3.77y_{12}\sin(y_{17}-y_{25}+2.02) - 3.36y_{12}y_{13}\sin(y_{26}-y_{25}+2.41) + 0.016 \\ g_{26} &= 10.7y_{13}^2 - 7.32y_{13}\sin(y_{17}-y_{26}+2.04) - 3.36y_{12}y_{13}\sin(y_{25}-y_{26}+2.41) \\ & \quad -2.58y_{13}y_{14}\sin(y_{27}-y_{26}+2.03) + 0.058 \\ g_{27} &= 5.34y_{14}^2 - 3.35y_9y_{14}\sin(y_{22}-y_{27}+2.01) - 2.58y_{13}y_{14}\sin(y_{26}-y_{27}+2.03) + 0.05. \end{cases}$$

3.7.2 Analysis

We consider a fault **occurring** at t_o, which consists of a power dropout in bus 1. This is modeled by a change in system dynamics and algebraic equations:

$$\begin{cases} \dot{x}_6 &= 15x_{11}\pi - \frac{3\pi(x_6-120\pi)}{5}, \\ g_1 &= 16.8y_1cos(y_{15}+1.89) - 1.0u_6 - 16.8y_1sin(y_{15}+1.89) + 25.5y_1^2 \\ & \quad +4.36y_1y_7cos(y_{20}+1.81) - 4.36y_1y_7sin(y_{20}+1.81) \\ g_2 &= 5.19cos(y_{16}-1.0y_{15}+1.8) - 1.0u_7 - 5.22y_2sin(x_2-1.0y_{15}) \\ & \quad +16.8y_1cos(y_{15}-1.89) + 5.71y_7cos(y_{20}-1.0y_{15}+1.89) \\ & \quad +5.63y_6cos(y_{19}-1.0y_{15}+1.89) + 10.6 \\ g_7 &= 4.56y_7cos(y_{17}-1.0y_{20}+1.57) - 1.0u_{12} + 5.71y_7cos(y_{15}-1.0y_{20}+1.89) + 9.57y_7^2 \\ & \quad +4.36y_1y_7cos(y_{20}-1.81) + 22.6y_6y_7cos(y_{19}-1.0y_{20}+1.88) + 0.076 \\ g_{15} &= 16.8y_1sin(y_{15}-1.89) - 5.22y_2cos(x_2-1.0y_{15}) - 5.19sin(y_{16}-1.0y_{15}+1.8) \\ & \quad -5.71y_7sin(y_{20}-1.0y_{15}+1.89) - 5.63y_6sin(y_{19}-1.0y_{15}+1.89) + 38.6 \\ g_{20} &= 35.5y_7^2 - 5.71y_7sin(y_{15}-1.0y_{20}+1.89) - 4.56y_7sin(y_{17}-1.0y_{20}+1.57) \\ & \quad +4.36y_1y_7sin(y_{20}-1.81) - 22.6y_6y_7sin(y_{19}-1.0y_{20}+1.88) + 0.016 \end{cases}$$

where all other states evolve by their previously mentioned dynamics. At time t_c the fault is **cleared**, which means the system behaves as given by the original dynamics. The verification goal is to check if the reachable set $\mathcal{R}(t)$ reverts back to the initial state after the fault is cleared, meaning

$$\exists t > t_c \colon \mathcal{R}(t) \subseteq \mathcal{R}(0).$$

The parameters for analysis are $t_o = 0.1$, $t_c = 0.13$, and time horizon $t_f = 5$. With $[-1,1]^p$ denoting the p-dimensional unit box, the initial set is given as $x^* + 0.01[-1,1]^5 \times 0.1[-1,1]^5 \times 0.001[-1,1]^5$ with

$$x^* = [0.33, -0.02, -0.22, -0.25, -0.23, 377.0, 377.0, 377.0, 377.0, 377.0, 2.0, 0.4, 0.0, 0.0, 0.0]^T$$

Furthermore, the piece-wise constant inputs are given as $u(t) = [2, 0.4, 0, \ldots, 0]^T$ for $t \in [0, t_o]$ and $t \in [t_c, t_f]$, and $u(t) = [0, 0.4, 0, \ldots, 0]^T$ for $t \in [t_o, t_c]$.

To simplify the analysis, we recommend to not treat the system as a hybrid system but to carry out three sequential analysis procedures for the system under normal operation, fault occurrence, and operation after fault.

3.8 Evaluation

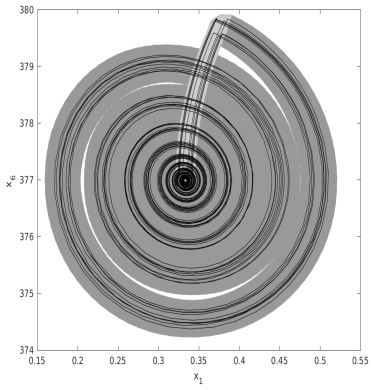
Successful verification of the safety specification is reported. Furthermore, a figure is provided in the (x_1, x_6) axes, with $x_1 \in [0.15, 0.55]$, and $x_6 \in [374, 380]$.

Table 7: Results of TSPS25 in terms of computation time.

tool	**computation time in [s]**
Ariadne	N/A
CORA	311
DynIbex	N/A
JuliaReach	N/A
KeYmaera X	N/A

3.8.1 Results

Only CORA is currently capable of addressing this benchmark. Visual results are shown in Fig. 7.



(a) CORA

Figure 7: Reachable set of the power system state in the x_1-x_6-plane for TSPS25.

Settings for Ariadne. We currently can only express differential-algebraic systems where each algebraic variable is assigned an expression in all other variables, therefore this benchmark can not be analyzed yet.

Settings for CORA. We employ an application-knowledge inspired system decomposition into two subsystems as was done in [7]. As performing the complete analysis on the subsystems would lead to a large approximation error due to the loss of information about the subsystems' coupling, we perform the general reachability analysis on the original system and use the decoupled systems solely to evaluate the Lagrange remainder for the linearization error [7]. CORA was run with a time step size of $\Delta t = 0.005$ for the normal operation, and with a time step size of $\Delta t = 0.001$ for operation under fault. For the normal operation after the fault, at time $t = 2$, we employ an enlarged time step of $\Delta t = 0.02$. At all times, we allowed a maximum zonotope order of 400.

Settings for DynIbex. DynIbex is capable of addressing this benchmark. However, computation times are very long because of the high dimensionality, and we did not manage to simulate during the whole time horizon with enough precision to obtain the correct result.

Settings for JuliaReach. JuliaReach currently has no approach to analyze differential-algebraic systems.

Settings for KeYmaera X. We could not address the benchmark at this time and will try next year; the high dimensionality of the system makes it challenging for the arithmetic solvers used by KeYmaera X, and manual arithmetic proofs are likely needed.

4 Conclusions and Outlook

This year, the competition confirmed the five participants from 2024.

Regarding benchmark evaluation, we introduced a new entry (TSPS25), that however for now could be addressed only by CORA. We additionally enhanced LOVO and ROBE to provide a clear verification objective in all cases.

The CVDP23 benchmark still proved a bit too difficult for Ariadne and was addressed by KeYmaera X in a simplified variant.

The TRAF22 benchmark was still not supported by Ariadne.

We like to mention that, triggered by the participation in this competition, individual tools made progress:

- CORA has started developing reachability algorithms incorporating splitting in the state space, as the CVDP23 benchmark is still too difficult for a single run. Preliminary tests on simpler dynamics have shown to be quite successful in automatically verifying reach-avoid specifications. Furthermore, we have noticed that one can exploit monotonicity in the range bounding of the derivatives in order to obtain tighter enclosures of the reachable set. This will be integrated in general in upcoming CORA releases.
- KeYmaera X added a tactic to certify Taylor models from the axioms of differential dynamic logic. This tactic was applied to the CVDP23 and LALO25 benchmarks, but is in need of performance improvements: it performs vastly more proof steps than other tactics for differential equation analysis (e.g., $54 \cdot 10^6$ proof steps to certify a Taylor model in CVDP23 vs 326 proof steps to analyze a differential invariant in ROBE25) and generates non-trivial proof obligations for the external arithmetic solvers used in KeYmaera X.

Summarizing, we believe that a benchmark suite with representative problems is of the utmost importance, in order to stimulate meaningful progress of all the participating tools. At the same time, we care about allowing all tools to solve all benchmarks and we will try to modify the most critical ones in order to achieve that. Consequently, for the next year we aim at refining the existing suite to advance in these directions, also possibly increasing the number of benchmarks.

5 Acknowledgments

Matthias Althoff and Maximilian Perschl acknowledge support by the German Research Foundation (DFG) project ConVeY under grant number GRK 2428.

Julien Alexandre dit Sandretto and Joris Tillet acknowledge support from CIEDS (STARTS project).

Luis Benet acknowledges support from PAPIIT-UNAM project IG-101122.

Christian Schilling acknowledges support from the Independent Research Fund Denmark under reference number 10.46540/3120-00041B and the Villum Investigator Grant S4OS under reference number 37819.

References

[1] Alessandro Abate, Matthias Althoff, Lei Bu, Gidon Ernst, Goran Frehse, Luca Geretti, Taylor T. Johnson, Claudio Menghi, Stefan Mitsch, Stefan Schupp, and Sadegh Soudjani. The ARCH-COMP friendly verification competition for continuous and hybrid systems. In *LNCS*, volume 14550, pages 1–37, 2025.

[2] Julien Alexandre dit Sandretto. Set-based b-series. *Mathematics*, 10(17), 2022. URL: https://www.mdpi.com/2227-7390/10/17/3165, doi:10.3390/math10173165.

[3] Julien Alexandre dit Sandretto and Alexandre Chapoutot. Validated Explicit and Implicit Runge-Kutta Methods. *Reliable Computing electronic edition*, 22, 2016.

[4] Julien Alexandre dit Sandretto and Alexandre Chapoutot. Validated Simulation of Differential Algebraic Equations with Runge-Kutta Methods. *Reliable Computing electronic edition*, 22, 2016.

[5] Julien Alexandre Dit Sandretto, Alexandre Chapoutot, and Olivier Mullier. Constraint-Based Framework for Reasoning with Differential Equations. In Çetin Kaya Koç, editor, *Cyber-Physical Systems Security*, pages 23–41. Springer International Publishing, December 2018.

[6] M. Althoff. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *Hybrid Systems: Computation and Control*, pages 173–182, 2013.

[7] M. Althoff. Formal and compositional analysis of power systems using reachable sets. *IEEE Transactions on Power Systems*, 29(5):2270–2280, 2014.

[8] M. Althoff. An introduction to CORA 2015. In *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 120–151, 2015.

[9] M. Althoff and D. Grebenyuk. Implementation of interval arithmetic in CORA 2016. In *Proc. of the 3rd International Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 91–105, 2016.

[10] M. Althoff, M. Koschi, and S. Manzinger. Commonroad: Composable benchmarks for motion planning on roads. In *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017.

[11] M. Althoff and B. H. Krogh. Zonotope bundles for the efficient computation of reachable sets. In *Proc. of the 50th IEEE Conference on Decision and Control*, pages 6814–6821, 2011.

[12] M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *Proc. of the 47th IEEE Conference on Decision and Control*, pages 4042–4048, 2008.

[13] Matthias Althoff. Benchmarks for the formal verification of power systems. In *Proc. of 9th International Workshop on Applied Verification of Continuous and Hybrid Systems*, 2022.

[14] Miguel Angel Barron. Stability of a ring of coupled van der Pol oscillators with non-uniform distribution of the coupling parameter. In *Journal of applied research and technology 14.1*, pages 62–66, 2016.

[15] Luis Benet and David P. Sanders. TaylorSeries.jl: Taylor expansions in one and several variables in Julia. *Journal of Open Source Software*, 4(36):1043, April 2019. doi:10.21105/joss.01043.

[16] Luis Benet and David P. Sanders. JuliaDiff/TaylorSeries.jl. https://github.com/JuliaDiff/TaylorSeries.jl, April 2021. doi:10.5281/zenodo.2601941.

[17] Luis Benet and David P. Sanders. JuliaIntervals/IntervalArithmetic.jl. https://github.com/JuliaIntervals/IntervalArithmetic.jl, May 2021. doi:10.5281/zenodo.3336308.

[18] Luis Benet and David P. Sanders. JuliaIntervals/TaylorModels.jl. https://github.com/JuliaIntervals/TaylorModels.jl, June 2021. doi:10.5281/zenodo.2613102.

[19] Luca Benvenuti, Davide Bresolin, Pieter Collins, Alberto Ferrari, Luca Geretti, and Tiziano Villa. Ariadne: Dominance checking of nonlinear hybrid automata using reachability analysis. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7550 LNCS:79 – 91, 2012.

[20] Carl Boettiger. An introduction to docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1):71–79, 2015.

[21] S. Bogomolov, M. Forets, G. Frehse, K. Potomkin, and C. Schilling. JuliaReach: a toolbox for set-based reachability. In *HSCC*, 2019. doi:10.1145/3302504.3311804.

[22] Sergiy Bogomolov, Marcelo Forets, Goran Frehse, Andreas Podelski, and Christian Schilling. Decomposing reach set computations with low-dimensional sets and high-dimensional matrices. *Inf. Comput.*, 2022. doi:10.1016/j.ic.2022.104937.

[23] Davide Bresolin, Pieter Collins, Luca Geretti, Roberto Segala, Tiziano Villa, and Sanja Živanović Gonzalez. A Computable and Compositional Semantics for Hybrid Automata. In *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, HSCC '20, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3365365.3382202.

[24] Davide Bresolin, Pieter Collins, Luca Geretti, Roberto Segala, Tiziano Villa, and Sanja Živanović Gonzalez. A computable and compositional semantics for hybrid systems. *Information and Computation*, 300, 2024. Cited by: 1. doi:10.1016/j.ic.2024.105189.

[25] Davide Bresolin, Luca Geretti, Riccardo Muradore, Paolo Fiorini, and Tiziano Villa. Formal verification applied to robotic surgery. *Lecture Notes in Control and Information Sciences*, 456:347 – 355, 2015. doi:10.1007/978-3-319-10407-2_40.

[26] Davide Bresolin, Luca Geretti, Riccardo Muradore, Paolo Fiorini, and Tiziano Villa. Formal verification of robotic surgery tasks by reachability analysis. *Microprocessors and Microsystems*, 39(8):836 – 842, 2015.

[27] N. Chan and S. Mitra. Verifying safety of an autonomous spacecraft rendezvous mission. In *ARCH17. 4th International Workshop on Applied Verification of Continuous and Hybrid Systems, collocated with Cyber-Physical Systems Week (CPSWeek) on April 17, 2017 in Pittsburgh, PA, USA*, pages 20–32, 2017. URL: http://www.easychair.org/publications/paper/342723.

[28] P. Collins, D. Bresolin, L. Geretti, and T. Villa. Computing the evolution of hybrid systems using rigorous function calculus. In *Proc. of the 4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS12)*, pages 284–290, Eindhoven, The Netherlands, June 2012.

[29] Vincent Drevelle and Jeremy Nicola. Vibes: A visualizer for intervals and boxes. *Mathematics in Computer Science*, 8(3):563–572, Sep 2014.

[30] Marcelo Forets and Christian Schilling. LazySets.jl: Scalable symbolic-numeric set computations. *Proceedings of the JuliaCon Conferences*, 1(1):11, 2021. doi:10.21105/jcon.00097.

[31] Nathan Fulton, Stefan Mitsch, Rose Bohrer, and André Platzer. Bellerophon: Tactical theorem proving for hybrid systems. In *Interactive Theorem Proving - 8th International Conference, ITP 2017, Brasília, Brazil, September 26-29, 2017, Proceedings*, pages 207–224, 2017. doi:10.1007/978-3-319-66107-0_14.

[32] Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, and André Platzer. Keymaera X: an axiomatic tactical theorem prover for hybrid systems. In *Automated Deduction - CADE-25 - 25th International Conference on Automated Deduction, Berlin, Germany, August 1-7, 2015, Proceedings*, pages 527–538, 2015. doi:10.1007/978-3-319-21401-6_36.

[33] James Gallicchio, Yong Kiam Tan, Stefan Mitsch, and André Platzer. Implicit definitions with differential equations for keymaera x (system description). In Jasmin Blanchette, Laura Kovacs, and Dirk Pattinson, editors, *IJCAR*, volume 13385 of *LNCS*. Springer, 2022. doi:10.1007/978-3-031-10769-6_42.

[34] L. Geretti, R. Muradore, D. Bresolin, P. Fiorini, and T. Villa. Parametric formal verification: the robotic paint spraying case study. In *Proceedings of the 20th IFAC World Congress*, pages 9248–9253, July 2017.

[35] F. Immler. Verified reachability analysis of continuous systems. In *Proc. of TACAS'15*, volume 9035 of *LNCS*, pages 37–51. Springer, 2015.

[36] N. Kochdumper and M. Althoff. Reachability analysis for hybrid systems with nonlinear guard sets. In *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, 2020.

[37] N. Kochdumper, P. Gassert, and M. Althoff. Verification of collision avoidance for CommonRoad traffic scenarios. In *Proc. of the 8th International Workshop on Applied Verification of Continuous and Hybrid Systems*, pages 184–194, 2021. doi:10.29007/1973.

[38] M. T. Laub and W. F. Loomis. A molecular network that produces spontaneous oscillations in excitable cells of dictyostelium. *Molecular Biology of the Cell*, 9:3521–3532, 1998.

[39] Stefan Mitsch. Implicit and explicit proof management in keymaera X. In *Proceedings of the 6th Workshop on Formal Integrated Development Environment, F-IDE@NFM 2021, Held online, 24-25th May 2021*, pages 53–67, 2021. doi:10.4204/EPTCS.338.8.

[40] Stefan Mitsch and André Platzer. The keymaera X proof IDE - concepts on usability in hybrid systems theorem proving. In *Proceedings of the Third Workshop on Formal Integrated Development Environment, F-IDE@FM 2016, Limassol, Cyprus, November 8, 2016*, pages 67–81, 2016. doi:10.4204/EPTCS.240.5.

[41] Stefan Mitsch and André Platzer. Modelplex: verified runtime validation of verified cyber-physical system models. *Formal Methods Syst. Des.*, 49(1-2):33–74, 2016. URL: https://doi.org/10.1007/s10703-016-0241-z, doi:10.1007/S10703-016-0241-Z.

[42] Stefan Mitsch and André Platzer. A retrospective on developing hybrid system provers in the keymaera family - A tale of three provers. In *Deductive Software Verification: Future Perspectives - Reflections on the Occasion of 20 Years of KeY*, pages 21–64. 2020. doi:10.1007/978-3-030-64354-6_2.

[43] Olivier Mullier, Alexandre Chapoutot, and Julien Alexandre dit Sandretto. Validated computation of the local truncation error of runge–kutta methods with automatic differentiation. *Optimization Methods and Software*, 33(4-6):718–728, 2018.

[44] Jorge A. Pérez-Hernández and Luis Benet. PerezHz/TaylorIntegration.jl. https://github.com/PerezHz/TaylorIntegration.jl, May 2021. doi:10.5281/zenodo.2562352.

[45] Saeed Peyghami, Pooya Davari, Mahmud Fotuhi-Firuzabad, and Frede Blaabjerg. Standard test systems for modern power system analysis: An overview. *IEEE industrial electronics magazine*, 13(4):86–105, 2019.

[46] André Platzer. A complete uniform substitution calculus for differential dynamic logic. *J. Autom. Reason.*, 59(2):219–265, 2017. doi:10.1007/s10817-016-9385-1.

[47] André Platzer and Yong Kiam Tan. Differential equation invariance axiomatization. *J. ACM*, 67(1):6:1–6:66, 2020. doi:10.1145/3380825.

[48] H. H. Robertson. The solution of a set of reaction rate equations. In *"Numerical analysis: an introduction"*, page 178–182. Academic Press, 1966.

[49] J. K. Scott, D. M. Raimondo, G. R. Marseglia, and R. D. Braatz. Constrained zonotopes: A new tool for set-based estimation and fault detection. *Automatica*, 69:126–136, 2016.

[50] Andrew Sogokon, Stefan Mitsch, Yong Kiam Tan, Katherine Cordwell, and André Platzer. Pegasus: sound continuous invariant generation. *Formal Methods Syst. Des.*, 58(1-2):5–41, 2021. doi:10.1007/s10703-020-00355-z.

[51] Yong Kiam Tan, Stefan Mitsch, and André Platzer. Verifying switched system stability with logic. In *HSCC '22: 25th ACM International Conference on Hybrid Systems: Computation and Control, Milan, Italy, May 4 - 6, 2022*, pages 2:1–2:11, 2022. doi:10.1145/3501710.3519541.

[52] R. Testylier and T. Dang. Nltoolbox: A library for reachability computation of nonlinear dynamical systems. In *Proc. of ATVA'13*, volume 8172 of *LNCS*, pages 469–473. Springer, 2013.

[53] K. Weihrauch. *Computable analysis*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2000.

[54] M. Wetzlinger, A. Kulmburg, A. Le Penven, and M. Althoff. Adaptive reachability algorithms for nonlinear systems using abstraction error analysis. *Nonlinear Analysis: Hybrid Systems*, 46:101252, 2022. URL: https://www.sciencedirect.com/science/article/pii/S1751570X22000607, doi:10.1016/j.nahs.2022.101252.