Linearization, Model Reduction and Reachability in Nonlinear ODEs

Michele Boreale, Luisa Collodi

Università degli Studi di Firenze
Dipartimento di Statistica, Informatica, Applicazioni (DiSIA)

October 17, 2022

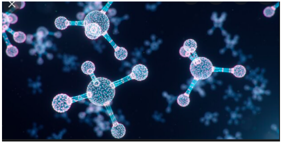
16th International Conference on Reachability Problems (RP'22)

Outline

1. Overview
2. Formalization
   - Linearization
   - Model-Order reduction
3. Global error bounding
4. Application to Reachability Analysis
5. Experiments
   - Graphical comparison
   - Reachsets: comparison with CORA and Flow*

Dynamical systems

- Mathematical models allow us to make predictions for natural phenomena, and are at the basis of the physical, chemical and biological sciences.
- Dynamical systems are a **fundamental modeling tool**, they represent the evolution of states in time of a certain phenomenon or device.
- Such systems often have a quite complex behavior and their analysis represents a challenging task and a very **active area of research**.
- We focus on purely **continuous systems**, which is the core problem, leaving the extension to hybrid systems to a future phase.



$$\begin{cases} \dot{x} = y \\ \dot{y} = \mu \cdot (1 - x^2) \cdot y - x \end{cases}$$

Van-der-Pol system (VDP)

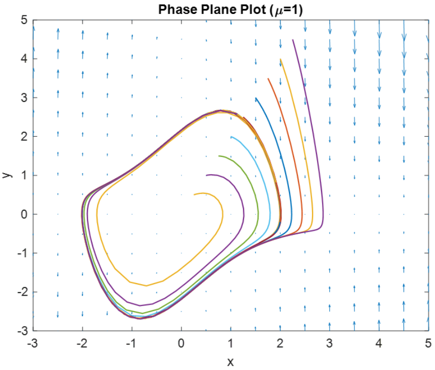
Nonlinear Ordinary Differential Equations (ODEs) will be analyzed.

For example, VDP models a system in which energy is added and subtracted, resulting in a periodic motion:

$$\begin{cases} \dot{x} = y \\ \dot{y} = \mu(1 - x^2)y - x \end{cases} \tag{1}$$

where:

- x: position coordinate.
- y: velocity of the motion.
- μ: scalar parameter, indicating the nonlinearity and the strength of the damping.



Overview

- Linear and Taylor approximations are fundamental tools in the analysis of nonlinear ODEs, but they generate approximate solutions that are typically only **locally accurate**.
- Our goal is to find conditions and methods to compute linear approximations of nonlinear ODEs that are **globally accurate**, as much as possible.
- **Carleman linearization** and **Krylov projection** are applied to produce a linear tractable system, approximating the original nonlinear system.
- **Error** w.r.t the original system is **bounded** over a infinite time horizon, under suitable stability conditions.

Formalization

Let

$$\dot{x} = f(x) \tag{2}$$

a continuous **system of ODEs**, with $x = (x_1, ..., x_n)^T \in \mathbb{R}^n$ a vector of *dependent variables*, $f = (f_1, ..., f_n)^T$ a *vector field* of locally Lipschitz analytic functions, and $x(t; x_0)$ its unique solution with initial condition $x(0) = x_0 \in X_0$.

$->$ We consider a function g that is real analytic, and we study the **observable** of system (2) via g, that is

$$\boxed{g \circ x(t; x_0)} \tag{3}$$

Formalization

- Let

$$\mathcal{L}_f(g) := \langle \nabla g \cdot f \rangle = \sum_{j=1}^{n} \frac{\partial g}{\partial x_j} \cdot f_j$$

the **Lie derivative** of g w.r.t. f.

- We fix a set $\mathcal{A} = \{\alpha_1, \alpha_2, ...\}$ of **basis functions** (e.g. monomials), and consider vector $\alpha = (\alpha_1, ..., \alpha_M)^T$ with $\alpha_i \in \mathcal{A}$ for $i = 1, ..., M$, then we assume that:

$$g = v^T \alpha,$$

where $v = (\lambda_1, ..., \lambda_M)^T \in \mathbb{R}^M$, and that:

$$\mathcal{L}_f(\alpha_i) = \sum_{j \geq 1} a_{ij} \alpha_j,$$

where $a_{ij} \in \mathbb{R}$, for $i = 1, .., M$, and $j \geq 1$.

- $M \in \mathbb{N}$ is chosen big enough to guarantee an approximation of $O(t^m)$ around $t = 0$, where m is the **order of approximation**.

Linearization (Carleman embedding)

- Given $A \in \mathbb{R}^{M\times M}$, $B \in \mathbb{R}^{M\times k}$, and $\psi = (\alpha_{M+1}, ..., \alpha_{M+k})^T$ a vector of monomials, where k is large enough to ensure that $a_{ij} = 0$ for each $j > M + k$ and $1 \le i \le M$:

$$\mathcal{L}_f(\alpha) = A\cdot\alpha + \boxed{B\cdot\psi} \longrightarrow \textit{remainder} \tag{4}$$

$$\frac{d}{dt}\alpha(x(t;x_0)) = \boxed{A}\cdot\alpha(x(t;x_0)) + \cancel{B\cdot\psi(x(t;x_0))} \longrightarrow \textit{first approximation layer} \tag{5}$$

- Thus, for any fixed initial condition $x_0 \in X_0$ of (2), we consider the following **linearized finite system** in variables $z = (z_1, ..., z_M)^T \in \mathbb{R}^m$

$$\dot{z} = Az \tag{6}$$

$$z(0) = \alpha(x_0) =: z_0 \tag{7}$$

whose *unique* solution is an approximation of $\alpha(x(t;x_0))$.

$->$ Matrix A is in general too large to be explicitly generated

Carleman Linearization for VDP

Let $m \in \mathbb{R}$ an **approximation order**, and $\alpha(x)$ the vector of monomials appearing in $\mathcal{L}^{(j)}(g)$, for $j = 0, ..., m-1$.

Considering VDP system, with $m = 2$, vector $\alpha(x)$ is given by:

$$\alpha(x) = (x_1, x_2, x_1^2 \cdot x_2)^T.$$

Thus, the **linearized finite system** becomes:

$$\dot{z} = \underbrace{\begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}}_{A} z$$

with $z \in \mathbb{R}^3$, and initial conditions $z(0) = \alpha(x_0) =: z_0$, for $x_0 = (x_{01}, x_{02})^T \in X_0$.

Model-order reduction

- We project A^T onto a subspace of $\mathbb{R}^M$ of dimension $m \ll M$, the m-dimensional **Krylov space** generated by v and A^T:

$$\mathcal{K}_m := span\{v, A^T v, (A^T)^2 v, ..., (A^T)^{m-1} v\} \tag{8}$$

- Thus, the following **reduced linear system** of size m in variables $y = (y_1, ..., y_m)^T$ is generated $\longrightarrow$ *second approximation layer*

$$\boxed{\begin{aligned} \dot{y} &= H_m^T y \qquad (9) \\ y(0) &= V^T z_0 =: y_0 \qquad (10) \end{aligned}}$$

with $H_m := V^T A^T V$, where $V = [v_1|...|v_m]$ is an orthonormal basis of $\mathcal{K}_m$.

- System (9) is used to approximate: $g(x(t)) \approx \boxed{\hat{g}(x(t)) := v^T V y(t; y_0)}$.
- The **error function** relative to g is given by: $\epsilon_g(t; x_0) := g(t; x_0) - \hat{g}(t; x_0)$. In particular, function $\epsilon_g(t; x_0)$ is $O(t^m)$ around $t = 0$, $\forall\, x_0 \in X_0$.

Global error bounding

Theorem

For any $t > 0$ such that $x(\tau; x_0)$ is defined for $\tau \in [0, t]$:

$$|\epsilon_g(t; x_0)| \leq ||v||_2 \int_0^t |h(x(\tau; x_0))| \cdot |(e^{(t-\tau)H_m^T})_{1,m}| \,\mathrm{d}\tau. \qquad (11)$$

If additionally H_m is **stable**, *then there is a constant $D > 0$ independent of t such that*

$$|\epsilon_g(t; x_0)| \leq ||v||_2 \, D \int_0^t |h(x(\tau; x_0))| \,\mathrm{d}\tau\,. \qquad (12)$$

where $h : \mathbb{R}^n \longrightarrow \mathbb{R}$ is the **remainder function**.

Application to Reachability Analysis

- Given a nonlinear system, we apply the outlined linearization scheme to compute an approximation of the flow, and bound the corresponding error to generate a **validated overapproximation** of the reachable set.
- Globally bounding the error for large t may be difficult, since it requires an upper bound of the solution $x(\tau; x_0)$ for $\tau \in [0, t]$. $->$ Not easy to find!
- The problem is solved breaking the interval $[0, t]$ into small **subintervals**: $0 = t_0, t_1, ..., t_N = t$, with $t_{k-1} < t_k$ for $k = 1, ..., N$, and using **local bounds** of the error function:

$$\gamma^-(t; S, E) \leq \epsilon_g(\tau; x_0) \leq \gamma^+(t; S, E)$$

with $\tau \in [0, t]$. γ^+, γ^- are obtained from the Taylor theorem in Lagrange form and using S, E, that are validated enclosures of $x(t; x_0)$ and $y(t; x_0)$, respectively.

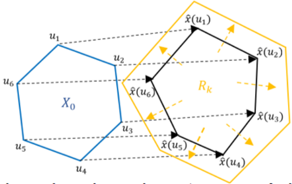
$->$ S, E can be computed through standard reachability analysis techniques.

Pseudocode

Algorithm 1 CKR

Input: vector field, initial set, order of approximation, $T \geq 0$, time horizon.
Output: a list of reachset.

```
set current vertices as the initial set vertices
generate the reduced linear system
solve the system to approximate the adv function
divide the time horizon into N sub-intervals
for k = 1, ..., N do
    compute enclosures in the k-subinterval for initial and reduced system variables
    advect current vertices and generate the new polytope
    compute bounds for the Lagrange remainder using enclosures
    inflate the generated polytope, and add it to reachsets list as k-th element
end for
```



Reachset overapproximation

Let $u_1, ..., u_p \in \mathbb{R}^n$ and $X_0 = \{\sum_{i=1}^{p} \lambda_i u_i : \lambda_i \geq 0 \textit{ and } \sum_{i=1}^{p} \lambda_i = 1\}$, in practice we will consider its halfspace representation: (C_0, b_0).

Definition (reachset overapproximation)

Let a sequence of vectors $\eta_k = (\eta_k^{(1)}, ..., \eta_k^{(\ell_k)})^T \in \mathbb{R}^{\ell_k}$ for $k = 0, 1, ...$ and of polytopes $R_k \subseteq \mathbb{R}^n$, that represent overapproximations of reachsets. Reachsets overapproximations can be defined inductively as: $\eta_0 := 0, R_0 := X_0$, and for $k \geq 1$:

$$\eta_k^{(j)} := \max_{\substack{\xi \in R_{k-1} \\ \delta \in [-\gamma_k, \gamma_k]}} c_j^T(\hat{x}(\Delta_k; \xi) + \delta) \qquad (j = 1, ..., \ell_k) \tag{13}$$

$$R_k := \{x \in \mathbb{R}^n : \; C_k x \leq \eta_k\}. \tag{14}$$

Reachset overapproximation

Let $u_1, \ldots, u_p \in \mathbb{R}^n$ and $X_0 = \{\sum_{i=1}^{p} \lambda_i u_i : \lambda_i \geq 0 \text{ and } \sum_{i=1}^{p} \lambda_i = 1\}$, in practice we will consider its halfspace representation: (C_0, b_0).

Definition (reachset overapproximation)

Let a sequence of vectors $\eta_k = (\eta_k^{(1)}, \ldots, \eta_k^{(\ell_k)})^T \in \mathbb{R}^{\ell_k}$ for $k = 0, 1, \ldots$ and of polytopes $R_k \subseteq \mathbb{R}^n$, that represent overapproximations of reachsets.
Reachsets overapproximations can be defined inductively as: $\eta_0 := 0$, $R_0 := X_0$, and for $k \geq 1$:

very demanding

$$\eta_k^{(j)} := \max_{\substack{\xi \in R_{k-1} \\ \delta \in [-\gamma_k, \gamma_k]}} c_j^T (\hat{x}(\Delta_k; \xi) + \delta) \qquad (j = 1, \ldots, \ell_k) \tag{13}$$

inspired by CheckMate

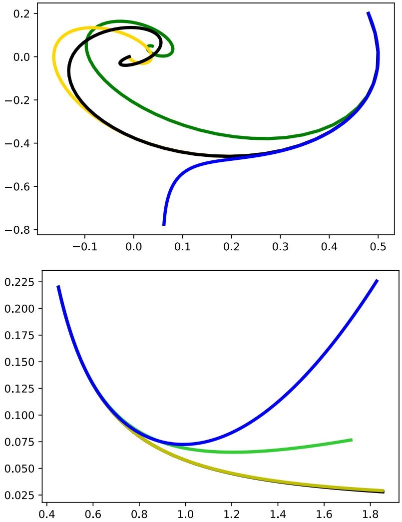
$$R_k := \{x \in \mathbb{R}^n : \; C_k x \leq \eta_k\}\,. \tag{14}$$

Experiments: graphical comparison

We applied a **proof-of-concept Python implementation** of our approximate solution method on a couple of systems in $\mathbb{R}^2$: Lotka Volterra and one taken from [2]. In detail, we show:

- exact solution computed numerically (yellow)
- our approximate solution (black)
- Taylor expansion of order $m-1$ of the solution from $t=0$ (blue)
- the solution of the linearized system (green).

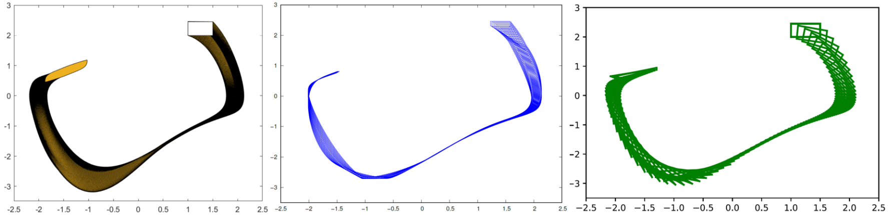
for $x_0 = (0.485, 0.2)^T$, and $t \in [0,1]$.



Experiments: reachability analysis

We apply a proof-of-concept Python implementation of CKR, and compare reachsets overapproximations generated by CKR and those produced by two state-of-the-art reachability tools: CORA, Flow*, on a few plain models.

Reachsets R_k for VDP obtained respectively from CORA, Flow* and CKR are:



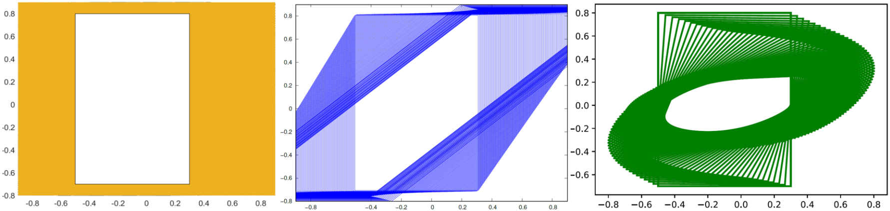
with $X_0 = [1.00, 1.50] \times [2.00, 2.45]$, $T = 5$, $m = 10$ for Flow* and CORA, and $m = 4$ for CKR.

Experiments: reachability analysis

We consider also the following system, that has the **origin as a stable equilibrium point**:

$$\begin{cases} \dot{x}_1 & = -x_1^3 + x_2 \\ \dot{x}_2 & = -x_1^3 - x_2^3 \end{cases} \tag{15}$$

Here we report the obtained reachsets for CORA, Flow* and CKR, with $X_0 = [-0.5, 0.3] \times [-0.7, 0.8]$.



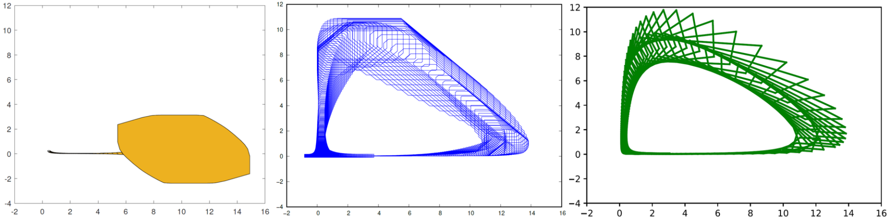
−> we have observed empirically that in this situation our method gives the best results

Experiments: reachability analysis

Finally, we analyze also the following unstable system:

$$\begin{cases} \dot{x}_1 & = x_1(1.5 - x_2) \\ \dot{x}_2 & = -x_2(3 - x_1) \end{cases} \tag{16}$$

The reachsets computed by CORA, Flow* and CKR, setting $X_0 = [-0.40, 0.52] \times [0.18, 0.27]$ are the following:



$->$ In general, relatively large initial sets have been considered.

Experiments: reachability analysis

- A **quantitative assessment of the accuracy** has been made. We measure accuracy as the average area of the reachsets returned by each algorithm until natural or premature termination:

$$av\_area = \frac{1}{N}\sum_{k=1}^{N} area(R_k) \tag{17}$$

- Also **times at which different algorithms stop** are compared, stopping may be due to a natural termination or to an explosion of the generated reachset (break down).

- **Execution times** are also compared, even if it makes little sense to compare a proof-of-concept implementation with highly optimized tools in this respect.

Experiments: reachability analysis

Sys	TH	Termination					Accuracy (average area)					Execution time				
		Flow*			CORA	CKR	Flow*			CORA	CKR	Flow*			CORA	CKR
		m=4	m=8	m=10	all m	all m	m=4	m=8	m=10	all m	m=4/5	m=4	m=8	m=10	m=4	m=4/5
(16)	1	**1**	**1**	**1**	**1**	**1**	0.02	0.02	0.02	**0.01**	**0.01**	**0.12**	1.45	3.90	0.17	13.31
	3	**3**	**3**	**3**	2.2*	**3**	22.75	6.20	6.18	1.27*	**2.82**	**0.98**	4.67	14.94	0.47*	50.31
	5	2.7*	**5**	**5**	2.2*	**5**	99.67*	4.37	4.35	1.27	**1.57**	2.74*	**8.55**	25.24	0.49*	94.79
(15)	1	**1**	**1**	**1**	0.6*	**1**	5.16	3.34	3.33	5.34*	**0.95**	**0.38**	6.92	23.06	4.28*	14.27
	3	1.3*	1.5*	1.5*	0.6*	**3**	8.37*	6.81*	6.10*	5.34*	**0.72**	5.04*	21.90*	67.76*	4.18*	**37.96**
	5	1.3*	1.5*	1.5*	0.6*	**5**	8.37*	6.81*	6.10*	5.34*	**0.62**	4.94*	19.84*	76.48*	5.08*	**64.42**
vdp	1	**1**	**1**	**1**	**1**	**1**	0.37	0.37	0.37	0.15	**0.12**	**0.13**	1.71	5.03	2.02	13.72
	3	**3**	**3**	**3**	**3**	**3**	0.16	0.15	0.15	0.09	**0.05**	**0.42**	5.05	15.42	5.13	37.05
	5	**5**	**5**	**5**	**5**	**5**	0.15	0.13	0.13	0.18	**0.07**	**0.77**	8.54	24.93	10.36	65.66

Conclusions

Contribution:

- A method to compute, given a nonlinear ODEs system, a linear system which is at the same time computationally tractable and useful to produce **globally accurate approximate solutions**.
- We have shown empirically that this method brings some benefit to classical reachability analysis in terms of accuracy.

For **future work**, it would be interesting:

- to investigate the relation of our method with other well-known linearization schemes, such as the **Koopman approach**.
- to explore the use of our reduced linearized system in **MPC**, and its application in **runtime monitoring**.

References

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

M. Boreale. Algorithms for exact and approximate linear abstractions of polynomial continuous systems. *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week), HSCC'18*, ACM, 2018.

X. Chen, E. Abraham, and S. Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In Proc. of CAV 2013: the 25th International Conference on Computer Aided Verification, vol. 8044 of LCNS, Springer, 2013.

R. M. Jungers, P. Tabuada. Non-local Linearization of Nonlinear Differential Equations via Polyflows. *2019 American Control Conference (ACC)*, pp. 1-6, doi: 10.23919/ACC.2019.8814337, 2019.

A. Mauroy, I. Mezic, Y. Susuki (eds.) *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*. Springer, 2020

Thank you
for your attention!