

Automatic Dynamic Parallelotope Bundles for Reachability of Nonlinear Dynamical Systems

Edward Kim

A thesis submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Science in the Department of Computer Science in the College of Arts and Science.

Chapel Hill
2022

Approved by:

Committee Member 1

Committee Member 2

Committee Member 3

Committee Member 4

Committee Member 5

Committee Member 6

©2022
Edward im
ALL RIGHTS RESERVED

ABSTRACT

Edward Kim: Automatic Dynamic Parallelotope Bundles for Reachability of Nonlinear Dynamical Systems

(Under the direction of Parasara Sridhar Duggirala)

Reachable set computation is an important technique for the verification of safety properties of dynamical systems. In this thesis, we investigate reachable set computation for discrete nonlinear systems based on parallelotope bundles. The crux of the reachability algorithm relies on computing an upper and lower bound on the supremum and infimum respectively of a nonlinear function over a rectangular domain. We cover two ways of computing these bounds: one method utilizing Bernstein polynomials and the other relying on a non-linear optimization tool developed by NASA, Kodiak. We aim to improve the traditional parallelotope-based reachability method by removing the manual step of parallelotope template selection in order to make the procedure fully automatic. Furthermore, we show that adding templates dynamically during computations can improve accuracy. To this end, we investigate two techniques for generating the template directions. The first technique approximates the dynamics as a linear transformation and generates templates using this linear transformation. The second technique uses Principal Component Analysis (PCA) of sample trajectories for generating templates. We have implemented our approach in a Python-based tool called Kaa. The tool is modular and use two types of global optimization solvers, the first using Bernstein polynomials and the second using the aforementioned Kodiak library. Additionally, we leverage the natural parallelism of the reachability algorithm and parallelize the Kaa implementation. Finally, we demonstrate the improved accuracy of our approach on several standard nonlinear benchmark systems, including a high-dimensional COVID19 model proposed by the Indian Supermodel Committee.

ACKNOWLEDGEMENTS

I would like to thank the UNC Computer Science community for supporting my work and providing me a warm community for pursuing this research. In particular, I would like to thank my advisor, Parasara Sridhar Duggirala, for the opportunity to perform research and participate in his group. Bineet Ghosh, Manish Goyal, Abel Karimi, and Meghan Stuart were all great colleagues and I enjoyed my time exchanging ideas, proofreading drafts, and socializing over group lunches. I would also like to thank Stanley Bak for helping me improve the implementation of Kaa and providing valuable feedback during this research. Juan Garcia made a wonderful colleague and an equally wonderful friend. The nights we raided Peabody and Phillips hall was certainly a memorable time. Finally, I would like to thank my parents and my family for their ardent, un-ending support.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
1 Introduction	1
1.1 Related Work	3
2 Preliminaries	4
2.1 Basic Definitions	4
2.2 Parallelotope-based Reachability	6
2.2.1 Parallelotopes	6
2.2.2 Bernstein Polynomials	7
2.2.3 The Static Algorithm	8
3 Dynamic Parallelotope Bundles	9
4 Experimental Results	10
4.0.1 Kaa	10
4.0.2 Benchmarks	10
4.0.3 COVID19 Supermodel	10
4.0.4 Comparison of Template Generation Techniques	10
BIBLIOGRAPHY	11

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

One of the most widely-used techniques for performing safety analysis of non-linear dynamical systems is reachable set computation. For example, reachability analysis has found many applications in formally verifying the safety properties of Cyber-physical Systems governed by Neural Network Controllers (Tran et al., 2019; Fan et al., 2020; Bak, 2021). The reachable set is defined to be the set of states visited by at least one of the trajectories of the system starting from an initial set and propagated forward in time by a finite fixed number of steps. Computing the exact reachable set for non-linear systems is challenging due to several reasons: First, unlike linear dynamical systems whose solutions can be expressed as closed form, non-linear dynamical systems generally do not admit such a nice form. Second, computationally speaking, current tools for performing non-linear reachability analysis are not very scalable. This is also in stark contrast to several scalable approaches developed for linear dynamical systems (Duggirala and Viswanathan, 2016; Bak and Duggirala, 2017). Finally, computing the reachable set using various set representations involves wrapping error which may be too conservative for practical use. That is, the overapproximation acquired at a given step would increase the conservativeness of the overapproximation for all future steps.

One of the several techniques for computing the overapproximation of reachable sets for discrete non-linear systems is to encode the reachable set through parallelotope bundles. Here, the reachable set is represented as a parallelotope bundle, a geometric data structure representing an intersection of several simpler objects called parallelotopes. One of the advantages of this technique is its exploitation of a special form of non-linear optimization problem to overapproximate the reachable set. The usage of a specific form of non-linear optimization mitigates many drawbacks involved with the scalability of non-linear analysis.

However, wrapping error still remains to be a problem for reachability using parallelotope bundles. An immediate reason stems from the responsibility of the practitioner to define the template directions specifying the parallelotopes. Often, these template directions are selected to be either the cardinal axis directions or some directions from octahedral domains. However, it is not certain that the axis-aligned and octagonal directions are optimal for computing reachable sets over general non-linear dynamics. Additionally, even an expert user of reachable set computation tools may not be able to ascertain a suitable set of template directions for computing reasonably accurate over-approximations of the reachable set. Picking unsuitable template directions would only cause the wrapping error to grow, leading to the aforementioned issue of overly conservative reachable sets.

In this thesis, we investigate techniques for generating template directions automatically and dynamically, which is the culmination of several publications in different venues (Kim and Duggirala, 2020; Kim et al., 2021; Geretti et al., 2021). Specifically, we propose a method where instead of the user providing the template directions to define the parallelotope bundle, he or she specifies the number of templates whose template directions are to be generated by our algorithm automatically.

To this end, we study two techniques for generating the said template directions. First, we compute a local linear approximation of the non-linear dynamics and use the linear approximation to compute the template directions. Second, we generate a set of trajectories sampled from within the reachable set and use Principal Component Analysis (PCA) over these trajectories. We observe that the accuracy of the reachable set can be drastically improved by using templates generated using these two techniques. To address scalability, we demonstrate that even when the size of the initial set increases, our template generation algorithm returns more accurate reachable sets than both manually-specified and random template directions. Finally, we experiment with our dynamic template generation algorithm's effectiveness on approximating the reachable set of high-dimensional COVID19 dynamics proposed by the Indian Supermodel Committee (National Supermodel Committee, 2020). The results were published in an ACM blogpost detailing the utility of reachable set computation in modeling disease dynamics (Bak et al., 2021).

1.1 Related Work

Reachable set computation of non-linear systems using template polyhedra and Bernstein polynomials has been first proposed in (Dang and Salinas, 2009). In (Dang and Salinas, 2009), Bernstein polynomial representation is used to compute an upper bound of a special type of non-linear optimization problem. This enclosing property of Bernstein polynomials has been actively studied in the area of global optimization (Nataray and Kotecha, 2002; Garloff, 2003; Nataraj and Arounassalame, 2007). Furthermore, several heuristics have been proposed for improving the computational performance of optimization using Bernstein polynomials (Smith, 2009; Muñoz and Narkawicz, 2013).

Several improvements to this algorithm were suggested in (Dang and Testylier, 2012; Sassi et al., 2012) and (Dang et al., 2014) extends it for performing parameter synthesis. The representation of parallelotope bundles for reachability was proposed in (Dreossi et al., 2016) and the effectiveness of using bundles for reachability was demonstrated in (Dreossi, 2017; Dreossi et al., 2017). However, all of these papers used static template directions for computing the reachable set. In other words, the user must specify the template directions before the reachable set computation proceeds.

Using template directions for reachable set has been proposed in (Sankaranarayanan et al., 2008) and later improved in (Dang and Gawlitza, 2011). Leveraging the principal component analysis of sample trajectories for computing reachable set has been proposed in (Stursberg and Krogh, 2003; Chen and Ábrahám, 2011; Seladji, 2017). More recently, connections between optimal template directions for reachability of linear dynamical systems and bilinear programming have been highlighted in (Gronski et al., 2019). For static template directions, octahedral domain directions (Clarisó and Cortadella, 2004) remain a popular choice.

CHAPTER 2

Preliminaries

We begin with some basic definitions pertaining to reachability and parallelotopes. The definition of Bernstein polynomials and the reachable set computation algorithm will be defined. Finally, an outline of the reachability algorithm given by (Dreossi et al., 2016) for polynomial dynamical systems will be presented.

2.1 Basic Definitions

As stated in the previous sections, this thesis pertains to the reachability analysis of dynamical systems. Roughly speaking, a dynamical system is governed by a set of differential equations such that the states of the system evolve according to the solutions of the said differential equations. The state of a system, denoted as x , lies in a domain $D \subseteq \mathbb{R}^n$ where the solution to the differential equations is defined. We restrict our attention to a specific definition of these dynamical systems:

Definition 2.1. A discrete-time nonlinear system is denoted as

$$x^+ = f(x) \tag{2.1}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a nonlinear function.

Intuitively, the function f takes input a state of the system and outputs the next step of the system evolved according to the non-linear dynamics. Here, the function f generally represents some discretized version of some specified continuous non-linear dynamical systems. Recall that a dynamical system is considered *linear* if its dynamics can be expressed as

$$x' = Ax, \quad A \in \mathbb{R}^{n \times n}$$

Otherwise, we deem the system to be *nonlinear*. Hence, in particular, the function f cannot be expressed as some matrix $A \in \mathbb{R}^{n \times n}$.

Examples of prominent non-linear dynamical systems include the Lotka-Volterra predator-prey model (Wangersky, 1978), FitzHugh Neuron model (FitzHugh, 1961), and recently introduced COVID19 disease model (National Supermodel Committee, 2020). Throughout this thesis, we discretize any continuous dynamics through the well-known Euler method. Thus, up to some error term of bounded degree, we can turn any non-linear system into the form given by Equation 2.1.

Example 2.1. The SIR Epidemic model is a 3-dimensional dynamical system governed by the following continuous dynamics:

$$\begin{aligned} s' &= \beta \cdot s_k i_k \\ i' &= \beta \cdot s_k i_k - \gamma \cdot i_k \\ r' &= \gamma \cdot i_k \end{aligned} \tag{2.2}$$

where s, i, r represent the fractions of a population of individuals designated as *susceptible*, *infected*, and *recovered* respectively. There are two parameters, namely β and γ , which influence the evolution of the system. β is labeled as the contraction rate and $1/\gamma$ is the mean infective period. Discretizing Equation 2.2 according the Euler method yields the dynamics:

$$\begin{aligned} s_{k+1} &= s_k - (\beta \cdot s_k i_k) \cdot \Delta \\ i_{k+1} &= i_k + (\beta \cdot s_k i_k - \gamma \cdot i_k) \cdot \Delta \\ r_{k+1} &= r_k + (\gamma \cdot i_k) \cdot \Delta \end{aligned} \tag{2.3}$$

Here, Δ is the discretization step and the index $k \in \mathbb{N}$ simply represents the current step. Note the non-linear terms $s_k i_k$ which precludes the expression of the dynamics as a linear transformation.

◇

The trajectory of a system that evolves according to Equation 2.1, denoted as $\xi(x_0)$ is a sequence x_0, x_1, \dots where $x_{i+1} = f(x_i)$. The k^{th} element in this sequence x_k is denoted as $\xi(x_0, k)$.

Definition 2.2. Given an initial set $\Theta \subseteq \mathbb{R}^n$, the *reachable set at step k* , denoted as Θ_k is defined as

$$\Theta_k = \{\xi(x, k) \mid x \in \Theta\} \quad (2.4)$$

If we set the number of steps to be some $n \in \mathbb{N}$, we say the *reachable set* is

$$\Theta = \bigcup_{i=1}^n \Theta_i \quad (2.5)$$

We will see in a future section an example of the reachable set of the discretized SIR model presented in Equation 2.3.

2.2 Parallelotope-based Reachability

2.2.1 Parallelotopes

A parallelotope P is a set of states in \mathbb{R}^n captured by the tuple $\langle \Lambda, c \rangle$ where $\Lambda \in \mathbb{R}^{2n \times n}$ is a matrix and $c \in \mathbb{R}^{2n}$ is a column vector. We impose the condition that $\Lambda_{i+n} = -\Lambda_i$ for all $i \in \{1, \dots, n\}$ such that

$$x \in P \text{ if and only if } \Lambda x \leq c.$$

We deem Λ as the *direction matrix* where Λ_i denotes the i^{th} row of Λ . The column vector c is called the *offset vector*. Alternatively, a parallelotope can also be represented in vertex-generator representation as $\langle v, g_1, \dots, g_n \rangle$. Here $v \in \mathbb{R}^n$ is called the *anchor* and $g_1, \dots, g_n, g_i \in \mathbb{R}^n$, are called the *generators*. The parallelotope is defined as the set:

$$P := \{x \mid \exists \alpha_1, \dots, \alpha_n \in [0, 1], x = v + \sum_{i=1}^n \alpha_i g_i\}$$

This representation is very similar to Zonotopes (Girard, 2005; Althoff et al., 2010) and Star sets (Dugirala and Viswanathan, 2016).

The vertex v_1 is obtained by solving the linear equation $\Lambda x = c_l$. The $j + 1$ vertex is obtained by solving the linear equation $\Lambda x = \mu_j$ where $\mu_j[i] = c_l[i]$ when $i \neq j$ and $\mu_j[j] = c_u[j]$. The anchor a of the parallelotope is the vertex v_1 and the generator $g_i = v_{i+1} - v_1$.

Notice that for a parallelotope P , the vertex-generator representation also defines the affine transformation that maps $[0, 1]^n$ to P . We denote this affine transformation as T_p .

Definition 2.3. A parallelotope bundle Q is a set of parallelotopes $\{P_1, \dots, P_m\}$ such that

$$Q = \bigcap_{i=1}^m P_i$$

.

2.2.2 Bernstein Polynomials

Given two multi-indices i and d of size n , where $i \leq d$, the Bernstein polynomial of degree d and index i is

$$\mathcal{B}_{i,d} = \beta_{i_1,d_1}(x_1)\beta_{i_2,d_2}(x_2) \dots \beta_{i_n,d_n}(x_n).$$

where $\beta_{i_m,d_m}(x_m) = \binom{d_m}{i_m} x_m^{i_m} (1 - x_m)^{d_m - i_m}$. Any polynomial function can be expressed in the Bernstein basis. The primary advantage of the Bernstein representation of a polynomial $h(x_1, \dots, x_n)$ is that an upper bound on the supremum and lower bound on the infimum of $h(x_1, \dots, x_n)$ in $[0, 1]^n$ can be computed purely by observing the coefficients of the polynomial in the Bernstein basis.

In other words, given a polynomial $h(x_1, \dots, x_n) = \sum_{j \in J} a_j \mathbf{x}_j$ where J is a set of multi-indices iterating through the degrees found in p with $a_j \in \mathbb{R}$, then $h(x_1, \dots, x_n)$ can be converted into its counterpart under the Bernstein basis, $h(x_1, \dots, x_n) = \sum_{j \in J} b_j \mathcal{B}_j$ where b_j are the corresponding Bernstein coefficients. The upper and lower bounds of $h(x_1, \dots, x_n)$ over $[0, 1]^n$ are bounded by the Bernstein coefficients:

$$\min\{b_1, \dots, b_m\} \leq \inf_{x \in [0,1]^n} h(x) \leq \sup_{x \in [0,1]^n} h(x) \leq \max\{b_1, \dots, b_m\}.$$

As mentioned earlier, a parallelotope P can also be represented as an affine transformation T_p from $[0, 1]^n$ to P . Therefore, upper bounds on the supremum of a function h over P is equivalent to upper bound of $h \circ T_p$ over $[0, 1]^n$. A similar argument follows for the lower bound on the infimum.

2.2.3 The Static Algorithm

CHAPTER 3

Dynamic Paralleotope Bundles

CHAPTER 4

Experimental Results

4.0.1 Kaa

4.0.2 Benchmarks

4.0.3 COVID19 Supermodel

4.0.4 Comparison of Template Generation Techniques

BIBLIOGRAPHY

- Althoff, M., Stursberg, O., and Buss, M. (2010). Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes. *Nonlinear analysis: hybrid systems*, 4(2):233–249.
- Bak, S. (2021). nnenum: Verification of relu neural networks with optimized abstraction refinement. In *NASA Formal Methods Symposium*, pages 19–36. Springer.
- Bak, S. and Duggirala, P. S. (2017). Simulation-equivalent reachability of large linear systems with inputs. In *International Conference on Computer Aided Verification*, pages 401–420. Springer.
- Bak, S., Kim, E., and Duggirala, P. S. (2021). Covid infection prediction using cps formal verification methods.
- Chen, X. and Ábrahám, E. (2011). Choice of directions for the approximation of reachable sets for hybrid systems. In *International Conference on Computer Aided Systems Theory*, pages 535–542. Springer.
- Clarísó, R. and Cortadella, J. (2004). The octahedron abstract domain. In *International Static Analysis Symposium*, pages 312–327. Springer.
- Dang, T., Dreossi, T., and Piazza, C. (2014). Parameter synthesis using parallelotopic enclosure and applications to epidemic models. In *International Workshop on Hybrid Systems Biology*, pages 67–82. Springer.
- Dang, T. and Gawlitza, T. M. (2011). Template-based unbounded time verification of affine hybrid automata. In *Asian Symposium on Programming Languages and Systems*, pages 34–49. Springer.
- Dang, T. and Salinas, D. (2009). Image computation for polynomial dynamical systems using the Bernstein expansion. In *International Conference on Computer Aided Verification*, pages 219–232. Springer.
- Dang, T. and Testylier, R. (2012). Reachability analysis for polynomial dynamical systems using the Bernstein expansion. *Reliable Computing*, 17(2):128–152.
- Dreossi, T. (2017). Sapo: Reachability computation and parameter synthesis of polynomial dynamical systems. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, pages 29–34.
- Dreossi, T., Dang, T., and Piazza, C. (2016). Parallelotope bundles for polynomial reachability. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pages 297–306.
- Dreossi, T., Dang, T., and Piazza, C. (2017). Reachability computation for polynomial dynamical systems. *Formal Methods in System Design*, 50(1):1–38.
- Duggirala, P. S. and Viswanathan, M. (2016). Parsimonious, simulation based verification of linear systems. In *International Conference on Computer Aided Verification*, pages 477–494. Springer.
- Fan, J., Huang, C., Chen, X., Li, W., and Zhu, Q. (2020). Reachnn*: A tool for reachability analysis of neural-network controlled systems. In *International Symposium on Automated Technology for Verification and Analysis*, pages 537–542. Springer.

- FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6):445–466.
- Garloff, J. (2003). The Bernstein expansion and its applications. *Journal of the American Romanian Academy*, 25:27.
- Geretti, L., Althoff, M., Benet, L., Chapoutot, A., Collins, P., Duggirala, P. S., Forets, M., Kim, E., Linares, U., et al. (2021). Arch-comp21 category report: Continuous and hybrid systems with nonlinear dynamics.
- Girard, A. (2005). Reachability of uncertain linear systems using zonotopes. In *International Workshop on Hybrid Systems: Computation and Control*, pages 291–305. Springer.
- Gronski, J., Sassi, M.-A. B., Becker, S., and Sankaranarayanan, S. (2019). Template polyhedra and bilinear optimization. *Formal Methods in System Design*, 54(1):27–63.
- Kim, E., Bak, S., and Duggirala, P. S. (2021). Automatic dynamic parallelotope bundles for reachability analysis of nonlinear systems. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 50–66. Springer.
- Kim, E. and Duggirala, P. S. (2020). Kaa: A Python implementation of reachable set computation using Bernstein polynomials. *EPiC Series in Computing*, 74:184–196.
- Muñoz, C. and Narkawicz, A. (2013). Formalization of Bernstein polynomials and applications to global optimization. *Journal of Automated Reasoning*, 51(2):151–196.
- Nataraj, P. S. and Arounassalame, M. (2007). A new subdivision algorithm for the Bernstein polynomial approach to global optimization. *International journal of automation and computing*, 4(4):342–352.
- Nataray, P. and Kotecha, K. (2002). An algorithm for global optimization using the Taylor–Bernstein form as inclusion function. *Journal of Global Optimization*, 24(4):417–436.
- National Supermodel Committee (2020). Indian Supermodel for Covid-19 Pandemic.
- Sankaranarayanan, S., Dang, T., and Ivančić, F. (2008). Symbolic model checking of hybrid systems using template polyhedra. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 188–202. Springer.
- Sassi, M. A. B., Testylier, R., Dang, T., and Girard, A. (2012). Reachability analysis of polynomial systems using linear programming relaxations. In *International Symposium on Automated Technology for Verification and Analysis*, pages 137–151. Springer.
- Seladji, Y. (2017). Finding relevant templates via the principal component analysis. In *International Conference on Verification, Model Checking, and Abstract Interpretation*, pages 483–499. Springer.
- Smith, A. P. (2009). Fast construction of constant bound functions for sparse polynomials. *Journal of Global Optimization*, 43(2-3):445–458.
- Stursberg, O. and Krogh, B. H. (2003). Efficient representation and computation of reachable sets for hybrid systems. In *International Workshop on Hybrid Systems: Computation and Control*, pages 482–497. Springer.

- Tran, H.-D., Manzananas Lopez, D., Musau, P., Yang, X., Nguyen, L. V., Xiang, W., and Johnson, T. T. (2019). Star-based reachability analysis of deep neural networks. In *International symposium on formal methods*, pages 670–686. Springer.
- Wangersky, P. J. (1978). Lotka-volterra population models. *Annual Review of Ecology and Systematics*, 9:189–218.