Revised CPA method to compute Lyapunov functions for nonlinear systems

Peter A. Giesl[a], Sigurdur F. Hafstein[b,*]

[a]Department of Mathematics, University of Sussex, Falmer BN1 9QH, United Kingdom
[b]School of Science and Engineering, Reykjavik University, Menntavegur 1, IS-101 Reykjavik, Iceland

Abstract

The CPA method uses linear programming to compute Continuous and Piecewise Affine Lyapunov function for nonlinear systems with asymptotically stable equilibria. In [14] it was shown that the method always succeeds in computing a CPA Lyapunov function for such a system. The size of the domain of the computed CPA Lyapunov function is only limited by the equilibrium's basin of attraction. However, for some systems, an arbitrary small neighborhood of the equilibrium had to be excluded from the domain a priori. This is necessary, if the equilibrium is not exponentially stable, because the existence of a CPA Lyapunov function in a neighborhood of the equilibrium is equivalent to its exponential stability as shown in [11]. However, if the equilibrium is exponentially stable, then this was an artifact of the method. In this paper we overcome this artifact by developing a revised CPA method. We show that this revised method is always able to compute a CPA Lyapunov function for a system with an exponentially stable equilibrium. The only conditions on the system are that it is C^2 and autonomous. The domain of the CPA Lyapunov function can be any a priori given compact neighborhood of the equilibrium which is contained in its basin of attraction. Whereas in a previous paper [10] we have shown these results for planar systems, in this paper we cover general n-dimensional systems.

Keywords: Lyapunov function, nonlinear system, exponential stability, basin of attraction, CPA function, piecewise linear function, linear programming

1. Introduction

Lyapunov functions, first introduced in [23], are a fundamental tool to determine the stability of equilibria and their regions of attraction. They can be used for very general systems, e.g. nonautonomous systems [22, 35, 16], arbitrary switched nonautonomous systems [15], or differential inclusions [5], but in this paper we concentrate on autonomous systems.

Consider the autonomous system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, $\mathbf{f} \in C^2(\mathbb{R}^n, \mathbb{R}^n)$, and assume that the origin is an exponentially stable equilibrium of the system. Denote by $\mathcal{A}$ its region of attraction. The standard method to verify the exponential stability of the origin is to solve the Lyapunov

*Corresponding author
Email addresses: p.a.giesl@sussex.ac.uk (Peter A. Giesl), sigurdurh@ru.is (Sigurdur F. Hafstein)

equation, i.e. to find a positive definite matrix $Q \in \mathbb{R}^{n\times n}$ that is a solution to $J^TQ + QJ = -P$, where $J := D\mathbf{f}(\mathbf{0})$ is the Jacobian of $\mathbf{f}$ at the origin and $P \in \mathbb{R}^{n\times n}$ is an arbitrary positive definite matrix. Then the function $\mathbf{x} \mapsto \mathbf{x}^TQ\mathbf{x}$ is a local Lyapunov function for the system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, i.e. it is a Lyapunov function for the system in some neighborhood of the origin, cf. e.g. Theorem 4.7 in [22]. The size of this neighborhood is a priori not known and is, except for linear $\mathbf{f}$, in general a poor estimate of $\mathcal{A}$, cf. [13]. This method to compute local Lyapunov functions is constructive because there is an algorithm to solve the Lyapunov equation that succeeds whenever it possesses a solution, cf. Bartels and Stewart [3].

The construction of Lyapunov functions for true nonlinear systems is a much harder problem than for linear systems. However, it has been studied intensively in the last decades and there have been numerous proposals of how to construct Lyapunov functions numerically. To name a few, Johansson and Rantzer proposed a construction method in [17] for piecewise quadratic Lyapunov functions for piecewise affine autonomous systems. In [7], Eghbal, Pariz, and Karimpour formulate the computation of piecewise quadratic Lyapunov functions for planar piecewise affine systems as linear matrix inequalities. In [32], Ratschan and She give an interval based branch-and-relax algorithm to compute polynomial Lyapunov-like functions for polynomial ODE. Another approach to numerically investigate the stability of nonlinear systems is, for example, given by Oishi in [27], where he considers the probabilistic computation of a stable control for systems that are parameter dependent, linear, and discrete. He uses a parameter dependent Lyapunov function.

Julian, Guivant, and Desages [20] and Julian [19] present a linear programming problem to construct piecewise affine Lyapunov functions for autonomous piecewise affine systems. This method can be used for autonomous, nonlinear systems if some a posteriori analysis of the generated Lyapunov function is done. In [18], Johansen uses linear programming to parameterize Lyapunov functions for autonomous nonlinear systems, but does not give error estimates. In [33], Rezaiee-Pajand and Moghaddasie proposed a different collocation method using two classes of basis functions. Giesl [8] proposed a method to construct Lyapunov functions for autonomous systems with an exponentially stable equilibrium by numerically solving a generalized Zubov equation, cf. [36]. A solution to Zubov's equation is a Lyapunov function for the system. He uses radial basis functions to approximate the solution and derives error estimates.

Parrilo [29] and Papachristodoulou and Prajna [28] consider the numerical construction of Lyapunov functions that can be expressed as sum of squares (SOS) of polynomials for autonomous polynomial systems. These ideas have been taken further by recent publications of Peet [30] and Peet and Papachristodoulou [31], where the existence of a polynomial Lyapunov function on bounded regions for exponentially stable systems is established. The Lyapunov functions are computed by means of convex optimization and are true Lyapunov functions and not approximations.

A complete Lyapunov functions, first introduced by Conley in [6], is a generalization of a Lyapunov function for compact invariant sets, as discussed here, to an object completely characterizing the decomposition of a flow into a chain-recurrent and a gradient-like part. Norton [26] even suggested that this characterization should be referred to as the Fundamental Theorem of Dynamical Systems. In [21], Kalies, Mischaikow and VanderVorst present an algorithmic approach to construct approximations to complete Lyapunov functions for discrete dynamical systems. By considering the time-T map of a continuous system, this method can be used to find an approximation to a complete Lyapunov function for a continuous dynamical system as well. In [2], Ban and Kalies implement this algorithm and give examples of computed Lyapunov functions.

In [25], Hafstein (alias Marinosson) presents a method to compute piecewise affine Lyapunov functions. In this method one first triangulates a compact neighborhood $\mathcal{C} \subset \mathcal{A}$ of the origin and then constructs a linear programming problem with the property that a continuous Lyapunov function V, affine on each n-simplex of the triangulation, i.e. a CPA Lyapunov function, can be constructed from any feasible solution to it. In [13] it was proved that for exponentially stable equilibria this method is always capable of generating a Lyapunov function $V : \mathcal{C} \setminus \mathcal{N} \longrightarrow \mathbb{R}$, where $\mathcal{N} \subset \mathcal{C}$ is an arbitrary small, a priori determined neighborhood of the origin. In [14], these results were generalized to asymptotically stable systems, in [15] to asymptotically stable, arbitrary switched, nonautonomous systems, and in [1] to asymptotically stable differential inclusions.

In [9], the authors showed that the triangulation scheme used in [25, 13, 14, 15] does in general generate suboptimal triangles at the equilibrium. However, in the same paper they proposed a new, fan-like triangulation around the equilibrium, and proved that a piecewise linear Lyapunov function with respect to this new triangulation always exists for planar systems. In [10], the authors showed how to compute a CPA Lyapunov function algorithmically for planar systems by using linear optimization. The modification to the algorithm in [15] is to use a fine, fan-like triangulation around the equilibrium, as suggested in [9]. The general n-dimensional case was treated in [11], where the authors proved, using different methods than in [9], that a piecewise linear Lyapunov function with respect to a modified, fan-like triangulation around the equilibrium always exists. However, the proof was non-constructive and it was not shown how to explicitly compute such a function. In this paper, the authors finish the work from [9, 10, 11] and deliver an algorithm to **compute** a CPA Lyapunov functions in n-dimensions and prove that the algorithm always succeeds in a finite number of steps whenever the system possesses an exponentially stable equilibrium.

The numerical discretization method presented in this paper is somewhat unusual since it is exact, i.e. it computes a true Lyapunov function and not an approximation. This is possible since a Lyapunov function is characterized through inequalities rather than equalities. Some other methods to construct Lyapunov functions, for example, the SOS method in [30, 31], also share this property. It should, however, be noted that the interplay between continuous systems and their discretization is very well understood. In particular, many important dynamical properties like attractors and basins of attraction are inherited by discretization, even for control systems. For a detailed discussion of this see the important work of Grüne [12].

Let us give an overview over the contents: In Section 2 we define a linear programming problem in Definition 6 and show that a solution of this problem parameterizes a CPA Lyapunov function in Theorem 1. In Section 3, we explain in Definition 17 how to algorithmically find a suitable triangulation for the linear programming problem from Definition 6. The main result is Theorem 5, showing that the algorithm from Definition 17 always succeeds in computing a CPA Lyapunov function for a system with an exponentially stable equilibrium. In Section 5 we give examples of CPA Lyapunov functions computed by our method. The paper ends with some concluding remarks in Section 6.

Notations

For a vector $\mathbf{x} \in \mathbb{R}^n$ and $p \geq 1$ we define the norm $\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$. We also define $\|\mathbf{x}\|_\infty = \max_{i\in\{1,\dots,n\}} |x_i|$. We will repeatedly use the Hölder inequality $|\mathbf{x}\cdot\mathbf{y}| \leq \|\mathbf{x}\|_p\|\mathbf{y}\|_q$, where $p^{-1} + q^{-1} = 1$, and the norm equivalence relations

$$\|\mathbf{x}\|_p \leq \|\mathbf{x}\|_q \leq n^{q^{-1}-p^{-1}}\|\mathbf{x}\|_p \quad \text{for } p > q.$$

The *induced matrix norm* $\|\cdot\|_p$ is defined by $\|A\|_p = \max_{\|\mathbf{x}\|_p=1} \|A\mathbf{x}\|_p$. Clearly $\|A\mathbf{x}\|_p \leq \|A\|_p\|\mathbf{x}\|_p$.

The *convex combination* of vectors $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^n$ is defined by $\text{co}\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_m\} := \{\sum_{i=0}^m \lambda_i\mathbf{x}_i : 0 \leq \lambda_i \leq 1, \sum_{i=0}^m \lambda_i = 1\}$. A set of vectors $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^n$ is called *affinely independent* if $\sum_{i=1}^m \lambda_i(\mathbf{x}_i - \mathbf{x}_0) = \mathbf{0}$ implies $\lambda_i = 0$ for all $i = 1, \dots, m$. This definition does not depend on the choice of $\mathbf{x}_0$. If $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^n$ are affinely independent, then the set $\text{co}\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_m\}$ is called an m-simplex. The set of r-times continuously differentiable functions from an open set $\mathcal{O} \subset \mathbb{R}^m$ to a set $\mathcal{P} \subset \mathbb{R}^n$ is denoted by $C^r(\mathcal{O}, \mathcal{P})$, i.e. all partial derivatives of order less than or equal to r of all components f_i of $\mathbf{f} \in C^r(\mathcal{O}, \mathcal{P})$ exist and are continuous. The preimage of a function f with respect to a subset $\mathcal{P}$ of its codomain is defined by $f^{-1}(\mathcal{P}) := \{x : f(x) \in \mathcal{P}\}$. We denote the closure of a set $\mathcal{N}$ by $\overline{\mathcal{N}}$ and the interior of $\mathcal{N}$ by $\mathcal{N}^\circ$. Finally, $\mathcal{B}_\delta$ is defined as the open $\|\cdot\|_2$-ball with center $\mathbf{0}$ and radius δ, i.e. $\mathcal{B}_\delta = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 < \delta\}$.

2. The linear programming problem

{programming}

Consider

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

where $\mathbf{f} \in C^2(\mathbb{R}^n, \mathbb{R}^n)$, $n \geq 2$, and $\mathbf{f}(\mathbf{0}) = \mathbf{0}$. It is well known that the asymptotic stability of the equilibrium at the origin is equivalent to the existence of a positive definite functional of the state space that is decreasing along the solution trajectories of the system, i.e. a continuously differentiable functional $V : \mathcal{O} \to \mathbb{R}$, where $\mathcal{O}$ is a connected open neighborhood of the origin, fulfilling $V(\mathbf{0}) = 0$ and $V(\mathbf{x}) > 0$ for all $\mathbf{x} \neq \mathbf{0}$ as well as

$$\nabla V(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) < 0 \;\text{ for all } \mathbf{x} \in \mathcal{O} \setminus \{\mathbf{0}\}. \tag{2}$$ {diffskil}

Such a functional V is called a (strict) Lyapunov function. It is also well known that the condition "continuously differentiable" can be mollified to "locally Lipschitz continuous" if the inequality (2) is replaced with

$$D^+V(\mathbf{x}) := \limsup_{h\to 0+} \frac{V(\mathbf{x} + h\mathbf{f}(\mathbf{x})) - V(\mathbf{x})}{h} < 0 \tag{3}$$ {cskil}

cf. e.g. Theorems 1.16 and 1.17 in [24].

In this paper we are interested in exponentially stable equilibria, i.e. the real parts of the eigenvalues of the Jacobian of $\mathbf{f}$ from (1) at the equilibrium at the origin are all strictly negative. We will show that if the origin is an exponentially stable equilibrium of (1), then a CPA Lyapunov function can be computed algorithmically by using linear programming.

Because we are only interested in an exponentially stable equilibrium at the origin, we only need to consider a specific type of Lyapunov functions that characterizes this kind of stability. Further, it is advantageous to define the set $\mathcal{N}$ of certain neighborhoods of the origin that we will repeatedly use in this paper. This is done in the next two definitions.

{neig}

Definition 1. Denote by $\mathcal{N}$ the set of all subsets $\mathcal{D} \subset \mathbb{R}^n$ that fulfill:

i) $\mathcal{D}$ is compact.
ii) The interior $\mathcal{D}^\circ$ of $\mathcal{D}$ is a connected open neighborhood of the origin.
iii) $\mathcal{D} = \overline{\mathcal{D}^\circ}$.

{def_lyap}

Definition 2. Consider the system (1). A Lipschitz continuous function $V : \mathcal{D} \to \mathbb{R}$, $\mathcal{D} \in \mathcal{N}$, is called a Lyapunov function for the system if $V(\mathbf{0}) = 0$ and there are constants $a, b, c > 0$ such that

$$a\|\mathbf{x}\|_2 \leq V(\mathbf{x}) \leq b\|\mathbf{x}\|_2 \text{ for all } \mathbf{x} \in \mathcal{D} \text{ and } \quad D^+V(\mathbf{x}) \leq -c\|\mathbf{x}\|_2 \text{ for all } \mathbf{x} \in \mathcal{D}^\circ.$$

Here, D^+V denotes the Dini derivative of V as defined in (3).

Remark 3. We will always refer to such a Lyapunov function as a “Lyapunov function in the sense of Definition 2”. Because $\mathcal{D}$ is compact, the concepts “Lipschitz continuous” and “locally Lipschitz continuous” coincide. Note that $D^+V(\mathbf{x})$ is not necessarily well defined for an $\mathbf{x}$ at the boundary of $\mathcal{D}$.

{sV}

Remark 4. If V is a Lyapunov function in the sense of Definition 2, then the Lyapunov function $V_s := sV$ with $s = \max\{a^{-1}, c^{-1}\}$ satisfies $\|\mathbf{x}\|_2 \leq V_s(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{D}$ and $D^+V_s(\mathbf{x}) \leq -\|\mathbf{x}\|_2$ for all $\mathbf{x} \in \mathcal{D}^\circ$.

Remark 5. The origin is an exponentially stable equilibrium of the system (1), if and only if it possesses a Lyapunov functions in the sense of Definition 2. In this case every connected component of a sublevel set $V^{-1}([0, r])$, $r > 0$, that is compact in $\mathcal{D}^\circ$, is a subset of the equilibrium's basin of attraction.

The “if” part follows e.g. from the Lyapunov function constructed in the proof of Theorem 4.7 in [22] and the fact that if V is a Lyapunov function, then so is $V^{\frac{1}{2}}$. The “only if” part follows e.g. from the proof of Theorem 4.10 in [22] for $a = 1$. Note that the theorem is not stated properly, because “V continuously differentiable” is contradictory to $k_1\|x\|^a \leq V(t, x) \leq k_2\|x\|^a$ if $0 < a \leq 1$. If, however, “V continuously differentiable” is replaced with “V locally Lipschitz in x”, then the proof is valid, even with $D^+V(\mathbf{x})$ defined as in (3).

We describe the idea of how to compute a Lyapunov function for the system (1), given a neighborhood $\mathcal{C} \in \mathcal{N}$ of the origin: we start by choosing a set $\mathcal{D} \in \mathcal{N}$, $\mathcal{D} \supset \mathcal{C}$, that can be triangulated, i.e. we can subdivide $\mathcal{D}$ into a set $\mathcal{T} := \{\mathfrak{S}_\nu : \nu = 1, 2, \ldots, N\}$ of n-simplices $\mathfrak{S}_\nu$, such that any two simplices in $\mathcal{T}$ intersect in a common face or are disjoint. Note that a face of an n-simplex is a k-simplex, $0 \leq k \leq n$, so this means that the intersection of two simplices $\mathfrak{S}_1$ and $\mathfrak{S}_2$ in $\mathcal{T}$ is either empty or $\mathfrak{S}_1 \cap \mathfrak{S}_2 = \text{co}\{\mathbf{x}_{i_0}, \mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_k}\}$, where $\mathbf{x}_{i_0}, \mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_k}$ are the common vertices of $\mathfrak{S}_1$ and $\mathfrak{S}_2$.

Then we construct a linear programming problem, of which every feasible solution parameterizes a CPA function V, i.e. a continuous function that is affine on each simplex in $\mathcal{T}$. Clearly such a function is Lipschitz continuous and for every $\mathfrak{S}_\nu \in \mathcal{T}$ the restriction of V to $\mathfrak{S}_\nu$ is given by $V|_{\mathfrak{S}_\nu}(\mathbf{x}) = \mathbf{w}_\nu \cdot \mathbf{x} + a_\nu$ with $\mathbf{w}_\nu \in \mathbb{R}^n$ and $a_\nu \in \mathbb{R}$. The linear programming problem imposes linear constraints that force $V(\mathbf{0}) = 0$ as well as the inequalities $V(\mathbf{x}) \geq \|\mathbf{x}\|_2$ for all $\mathbf{x} \in \mathcal{D}$ and $\mathbf{w}_\nu \cdot \mathbf{f}(\mathbf{x}) \leq -\|\mathbf{x}\|_2$ for every $\nu = 1, 2, \ldots, N$ and every $\mathbf{x} \in \mathfrak{S}_\nu$. Since we cannot use a linear programming problem to check the conditions $V(\mathbf{x}) \geq \|\mathbf{x}\|_2$ and $\mathbf{w}_\nu \cdot \mathbf{f}(\mathbf{x}) \leq -\|\mathbf{x}\|_2$ for more than finitely many $\mathbf{x}$, the essence of the linear programming problem is how to ensure the inequalities by only using a finite number of points in $\mathcal{D}$. Note that the condition $\mathbf{w}_\nu \cdot \mathbf{f}(\mathbf{x}) \leq -\|\mathbf{x}\|_2$ is equivalent to (3) for our specific choice of V as shown later.

First, one verifies that if $\mathfrak{S}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$, then it is enough to force $V(\mathbf{x}_i) \geq \|\mathbf{x}_i\|_2$, $i = 0, 1, \ldots, n$, to ensure that $V(\mathbf{x}) \geq \|\mathbf{x}\|_2$ for all $\mathbf{x} \in \mathfrak{S}_\nu$.

Second, for every $\mathfrak{S}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$ one picks one vertex, say $\mathbf{x}_0$, and introduces positive constants $E_{\nu,i}$, $i = 1, 2, \ldots, n$, dependent on the vector field $\mathbf{f}$ and the n-simplex $\mathfrak{S}_\nu$, and then uses the linear programming problem to force $\mathbf{w}_\nu \cdot \mathbf{f}(\mathbf{x}_0) \leq -\|\mathbf{x}_0\|_2$ and $\mathbf{w}_\nu \cdot \mathbf{f}(\mathbf{x}_i) + E_{\nu,i}\|\mathbf{w}_\nu\|_1 \leq -\|\mathbf{x}_i\|_2$ for $i = 1, 2, \ldots, n$. For practical reasons, it is convenient to introduce the constants $E_{\nu,0} := 0$ for $\nu = 1, 2, \ldots, N$. Then the last two inequalities can be combined to

$$\mathbf{w}_\nu \cdot \mathbf{f}(\mathbf{x}_i) + E_{\nu,i}\|\mathbf{w}_\nu\|_1 \leq -\|\mathbf{x}_i\|_2 \quad \text{for} \quad i = 0, 1, \ldots, n.$$

These last inequalities can be made linear in the components of $\mathbf{w}_\nu$, and with a proper choice of the $E_{\nu,i}$ they ensure that $\mathbf{w}_\nu \cdot \mathbf{f}(\mathbf{x}) \leq -\|\mathbf{x}\|_2$ for all $\mathbf{x} \in \mathfrak{S}_\nu$. Since this holds true for every $\mathfrak{S}_\nu \in \mathcal{T}$, one can show that $D^+V(\mathbf{x}) \leq -\|\mathbf{x}\|_2$ for all $\mathbf{x} \in \mathcal{D}^\circ$. Hence, e.g. by Theorem 2.16 in [15], the function $V : \mathcal{D} \to \mathbb{R}$ is a Lyapunov function for the system (1) in the sense of Definition 2 and so is V restricted to $\mathcal{C}$.
We now state our linear programming problem for the system (1) and prove that any feasible solution parameterizes a Lyapunov function for the system. The linear programming problem is defined in the next definition. It is followed by several explanatory and clarifying remarks.

{LPprob}

Definition 6 (The linear programming problem). The variables of the linear programming problem for the system (1) are $V_{\mathbf{x}}$ for all vertices $\mathbf{x}$ of the n-simplices of the triangulation $\mathcal{T}$ defined in Step 1 and $C_{\nu,i}$, $i = 1, 2, \ldots, n$, for every $\mathfrak{S}_\nu \in \mathcal{T}$. The constraints of the linear programming problem are given by (4), (5), and (6). The construction of the linear programming problem is as follows:

1. We triangulate $\mathcal{D} \in \mathcal{N}$ into a finite number of n-simplices $\mathcal{T} = \{\mathfrak{S}_\nu : \nu = 1, 2, \ldots, N\}$. That is $\mathcal{D} := \bigcup_{\mathfrak{S}_\nu \in \mathcal{T}} \mathfrak{S}_\nu$ and any two different simplices from $\mathcal{T}$ intersect in a common face or not at all. Further, we demand from our triangulation that whenever $\mathbf{0} \in \mathfrak{S}_\nu \in \mathcal{T}$, then $\mathbf{0}$ is a vertex of $\mathfrak{S}_\nu$.
   We define $V : \mathcal{D} \to \mathbb{R}$ uniquely by:
   - $V : \mathcal{D} \to \mathbb{R}$ is continuous.

- For every n-simplex $\mathfrak{S}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{T}$ we have $V(\mathbf{x}_i) = V_{\mathbf{x}_i}$, $i = 0, 1, \ldots, n$, and the restriction of V to any n-simplex $\mathfrak{S}_\nu \in \mathcal{T}$ is affine, i.e. there is a $\mathbf{w}_\nu \in \mathbb{R}^n$ and an $a_\nu \in \mathbb{R}$ such that $V(\mathbf{x}) = \mathbf{w}_\nu \cdot \mathbf{x} + a_\nu$ for every $\mathbf{x} \in \mathfrak{S}_\nu$.

For such a function we define $\nabla V_\nu := \mathbf{w}_\nu$ for $\nu = 1, 2, \ldots, N$. The components of the vector ∇V_ν are linear in $V_{\mathbf{x}_0}, V_{\mathbf{x}_1}, \ldots, V_{\mathbf{x}_n}$, where $\mathfrak{S}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$, cf. Remark 9.

2. We set $V_{\mathbf{0}} = 0$. For every $\mathfrak{S}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{T}$ and every vertex $\mathbf{x}_i \neq \mathbf{0}$

$$V_{\mathbf{x}_i} \geq \|\mathbf{x}_i\|_2 \tag{4}$$ {LC1}

is a linear constraint of the problem.

3. For every $\mathfrak{S}_\nu \in \mathcal{T}$ and $i = 1, 2, \ldots, n$

$$|\nabla V_{\nu,i}| \leq C_{\nu,i}, \tag{5}$$ {LC3}

where $\nabla V_{\nu,i}$ is the i-th component of the vector ∇V_ν, is a linear constraint of the problem, cf. Remark 10.

4. For every $\mathfrak{S}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{T}$ and every vertex $\mathbf{x}_i \in \mathfrak{S}_\nu$, $i = 0, 1, \ldots, n$,

$$-\|\mathbf{x}_i\|_2 \geq \nabla V_\nu \cdot \mathbf{f}(\mathbf{x}_i) + E_{\nu,i} \sum_{j=1}^{n} C_{\nu,j}, \tag{6}$$ {LC4}

is a linear constraint of the problem. In this inequality

$$E_{\nu,i} := \frac{nB_\nu}{2} \|\mathbf{x}_i - \mathbf{x}_0\|_2 \left(\max_{j=1,2,\ldots,n} \|\mathbf{x}_j - \mathbf{x}_0\|_2 + \|\mathbf{x}_i - \mathbf{x}_0\|_2 \right), \tag{7}$$ {Edef}

where B_ν is a constant fulfilling

$$B_\nu \geq \max_{m,r,s=1,2,\ldots,n} \max_{\mathbf{z} \in \mathfrak{S}_\nu} \left| \frac{\partial^2 f_m}{\partial x_r \partial x_s}(\mathbf{z}) \right|.$$

See Remark 11 on these constants.
If $\mathbf{0} \notin \mathfrak{S}_\nu$, then we can choose the vertex $\mathbf{x}_0$ arbitrarily. If $\mathbf{0} \in \mathfrak{S}_\nu$, then $\mathbf{0}$ is necessarily a vertex of $\mathfrak{S}_\nu$, and in this case we set $\mathbf{x}_0 = \mathbf{0}$.

Remark 7. An explicit triangulation as in Step 1 is constructed in Definition 13.

Remark 8. It is not necessary to force $V(\mathbf{x}) \leq b\|\mathbf{x}\|_2$ for all $\mathbf{x} \in \mathcal{D}$ explicitly with linear constraints because this inequality is automatically fulfilled with $b := \max_{\mathfrak{S}_\nu \in \mathcal{T}} \|\nabla V_\nu\|_2$.

{nVrem}

Remark 9. The components of the vector ∇V_ν are linear in $V_{\mathbf{x}_0}, V_{\mathbf{x}_1}, \ldots, V_{\mathbf{x}_n}$, where $\mathfrak{S}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$. To see this, define the matrix X_ν by writing the components of the vector $\mathbf{x}_i - \mathbf{x}_0$ in its i-th row, $i = 1, 2, \ldots, n$. Since $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n$ are affinely independent, the matrix X_ν is invertible. Define the column vector $\mathbf{v}_\nu^*$ by setting $V_{\mathbf{x}_i} - V_{\mathbf{x}_0}$ as its i-th component for $i = 1, 2, \ldots, n$. Now $V(\mathbf{x}) = \nabla V_\nu \cdot \mathbf{x} + a_\nu$ for every $\mathbf{x} \in \mathfrak{S}_\nu$ where $\nabla V_\nu = X_\nu^{-1}\mathbf{v}_\nu^*$. Indeed,

$$X_\nu^{-1}\mathbf{v}_\nu^* \cdot (\mathbf{x}_i - \mathbf{x}_0) = (\mathbf{x}_i - \mathbf{x}_0)^T X_\nu^{-1}\mathbf{v}_\nu^* = \mathbf{e}_i^T \mathbf{v}_\nu^* = V(\mathbf{x}_i) - V(\mathbf{x}_0) = \nabla V_\nu \cdot (\mathbf{x}_i - \mathbf{x}_0)$$

for $i = 1, 2, \ldots, n$.

{nVrem2}

Remark 10. By Remark 9, the components of the vector ∇V_ν are linear in $V_{\mathbf{x}_0}, V_{\mathbf{x}_1}, \ldots, V_{\mathbf{x}_n}$ and, e.g. by [4, p. 17], this implies that the constraints (5) are linear in the variables $V_{\mathbf{x}_0}, V_{\mathbf{x}_1}, \ldots, V_{\mathbf{x}_n}$, too.

{Bdissc}

Remark 11. The constants B_ν in Step 4 are the only parameters of the linear programming problem that are not computed algorithmically. However, one only needs to obtain some **upper bounds** on the second-order partial derivatives of $\mathbf{f}$ and because these bounds do not have to be close, this is usually a very simple task.

Remark 12. We explain the choice of the vertex $\mathbf{x}_0$ in Step 4: If $\mathbf{0} \in \mathfrak{S}_\nu$ then $\mathbf{0}$ is necessarily a vertex of $\mathfrak{S}_\nu$ and in this case we must set $\mathbf{x}_0 = \mathbf{0}$, for otherwise the constraint (6) could not be fulfilled if $B_\nu > 0$. To verify this, observe that if e.g. $\mathbf{x}_1 = \mathbf{0}$ and then $\mathbf{x}_0 \neq \mathbf{0}$ we have

$$0 = -\|\mathbf{x}_1\|_2 \geq \nabla V_\nu \cdot \underbrace{\mathbf{f}(\mathbf{x}_1)}_{=\mathbf{0}} + E_{\nu,1} \sum_{j=1}^{n} C_{\nu,j} = E_{\nu,1} \sum_{j=1}^{n} C_{\nu,j}.$$

But we have by (7)

$$E_{\nu,1} := \frac{nB_\nu}{2} \|\mathbf{x}_1 - \mathbf{x}_0\|_2 \left(\max_{j=1,2,\ldots,n} \|\mathbf{x}_j - \mathbf{x}_0\|_2 + \|\mathbf{x}_1 - \mathbf{x}_0\|_2 \right) > 0$$

so (6) cannot be fulfilled unless $\sum_{j=1}^{n} C_{\nu,j} = 0$, which is impossible because by (5), V would be constant on $\mathfrak{S}_\nu$ and (6) could not be fulfilled for all vertices of $\mathfrak{S}_\nu$.

However, as we set $\mathbf{x}_0 = \mathbf{0}$, we have

$$E_{\nu,0} := \frac{nB_\nu}{2} \|\mathbf{x}_0 - \mathbf{x}_0\|_2 \left(\max_{j=1,2,\ldots,n} \|\mathbf{x}_j - \mathbf{x}_0\|_2 + \|\mathbf{x}_0 - \mathbf{x}_0\|_2 \right) = 0$$

and (6) is trivially fulfilled. Obviously there is no loss of generality.

If $\mathbf{0} \notin \mathfrak{S}_\nu$, then we can choose $\mathbf{x}_0$ arbitrarily. Different choices will obviously lead to different linear programming problems, but they are all equivalent in the sense that a CPA Lyapunov function can be parameterized from a feasible solution to any of them, cf. Theorem 1.

If a linear programming problem from Definition 6 possesses a feasible solution, i.e. the variables $V_{\mathbf{x}_i}$ and $C_{\nu,i}$ have values such that the constraints (4), (5), and (6) are all fulfilled, then it is always possible to algorithmically find a feasible solution, e.g. by the simplex algorithm. In this case, the function $V : \mathcal{D} \to \mathbb{R}$ defined in Definition 6 is a CPA Lyapunov function in the sense of Definition 2 for the system (1) by the next theorem.

{LPtheo}

Theorem 1. *Assume that a linear programming problem from Definition 6 has a feasible solution and let $V : \mathcal{D} \to \mathbb{R}$ be the CPA function parameterized by it. Then V is a Lyapunov function in the sense of Definition 2 for the system (1) used in the construction of the linear programming problem.*

Proof: This result was proved for $n = 2$ in Theorem 4.6 in [10] with a similar notation. The proof for a general $n \geq 2$ is identical, just replace 2 by n, where appropriate. □

3. The Algorithm

{algorithm}

We define a parameterized set of triangulations $\mathcal{T}_{K,b}^{\mathcal{C}}$ in Definition 13 of a superset of any $\mathcal{C} \in \mathcal{N}$. A superset is necessary because not all $\mathcal{C} \in \mathcal{N}$ can be triangulated. We explain the choice of the superset in Definition 13 below and define our algorithm in Definition 17.

3.1. Triangulation

{tria}

For the algorithm to construct a piecewise affine Lyapunov function we need to fix our triangulation. That is a subdivision of $\mathbb{R}^n$ into n-simplices, such that the intersection of any two different simplices is either empty or a k-simplex, $0 \leq k < n$, and then its vertices are the common vertices of the two different n-simplices. Such a structure is often referred to as a simplicial n-complex.

We do this by extending the local simplicial n-complex from [11]; this is similar to extending the local planar triangulation from [9] in [10]. The main idea is to take the intersection of the boundary of a hypercube $[-b, b]^n$, $b > 0$, with the simplices in a simplicial n-complex as in [15], such that the boundary is subdivided into a simplicial $(n-1)$-complex. The simplicial n-complex must be chosen such that the boundary of the hypercube $[-b, b]^n$ and a simplex in complex can only intersect in a face of the simplex. Then the intersection is naturally a simplicial $(n-1)$-complex itself and we then add the origin as a vertex to all the simplices in this complex to get a new simplicial n-complex locally at the origin. We refer to this local triangulation as the *simplicial fan* because of its similarity to the 3D graphics primitive triangle fan. Outside of the hypercube $[-b, b]^n$ we continue to use the simplicial n-complex from [15].

For the construction we use the set S_n of all permutations of the numbers $1, 2, \ldots, n$, the characteristic functions $\chi_{\mathcal{J}}(i)$ equal to one if $i \in \mathcal{J}$ and equal to zero if $i \notin \mathcal{J}$, and the standard orthonormal basis $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$ of $\mathbb{R}^n$. Further, we use the functions $\mathbf{R}^{\mathcal{J}} : \mathbb{R}^n \to \mathbb{R}^n$, defined for every $\mathcal{J} \subset \{1, 2, \ldots, n\}$ by

$$\mathbf{R}^{\mathcal{J}}(\mathbf{x}) := \sum_{i=1}^{n} (-1)^{\chi_{\mathcal{J}}(i)} x_i \mathbf{e}_i.$$

Thus $\mathbf{R}^{\mathcal{J}}(\mathbf{x})$ puts a minus in front of the coordinate x_i of $\mathbf{x}$ if $i \in \mathcal{J}$.

Note that the two parameters b and K of the triangulations $\mathcal{T}_{K,b}^{\mathcal{C}}$ and $\mathcal{T}_{K,b}$ refer to the size of the hypercube $[-b, b]^n$ covered by its simplicial fan at the origin and to the fineness, respectively. $\mathcal{C}$ refers to the a priori given compact neighborhood $\mathcal{C} \in \mathcal{N}$ of the origin.

{triconstr}

Definition 13. Let $\mathcal{C} \in \mathcal{N}$ be a given subset of $\mathbb{R}^n$. We will define a triangulation $\mathcal{T}_{K,b}^{\mathcal{C}}$ of a $\mathcal{D} \in \mathcal{N}$, $\mathcal{D} \supset \mathcal{C}$, that approximates $\mathcal{C}$. To construct the triangulation $\mathcal{T}_{K,b}^{\mathcal{C}}$, we first define the triangulations $\mathcal{T}^{\text{std}}$, $\mathcal{T}_K^{\text{std}}$, and $\mathcal{T}_{K,b}^{\text{std}}$ as intermediate steps.

1. The standard triangulation $\mathcal{T}^{\text{std}}$ consists of the simplices

$$\mathfrak{S}_{\mathbf{z},\mathcal{J},\sigma} := \text{co}\left\{ \mathbf{R}^{\mathcal{J}}\left(\mathbf{z} + \sum_{i=1}^{j} \mathbf{e}_{\sigma(i)} \right) \,:\, j = 0, 1, 2, \ldots, n \right\}$$

for all $\mathbf{z} \in \mathbb{N}_0^n$, all $\mathcal{J} \subset \{1, 2, \ldots, n\}$, and all $\sigma \in S_n$.

2. Choose a $K \in \mathbb{N}_0$ and consider the intersections of the n-simplices $\mathfrak{S}_{\mathbf{z},\mathcal{J},\sigma}$ in $\mathcal{T}^{\rm std}$ and the boundary $[-2^K, 2^K]^n$. We are only interested in those intersections that are $(n-1)$-simplices, i.e. we take every simplex with vertices $\mathbf{x}_j := \mathbf{R}^{\mathcal{J}}\left(\mathbf{z} + \sum_{i=1}^{j} \mathbf{e}_{\sigma(i)}\right)$, $j \in \{0, 1, \ldots, n\}$, where exactly one vertex satisfies $\|\mathbf{x}_{j^*}\|_\infty \neq 2^K$ and the other n of the $n+1$ vertices satisfy $\|\mathbf{x}_j\|_\infty = 2^K$ for $j \in \{0, 1, \ldots, n\} \setminus \{j^*\}$. Then we replace the vertex $\mathbf{x}_{j^*}$ by $\mathbf{0}$.
   Thus, we obtain a new triangulation of $[-2^K, 2^K]^n$, which is denoted by $\mathcal{T}_K^{\rm std}$.
3. Now choose a constant $b > 0$ and scale down the triangulation $\mathcal{T}_K^{\rm std}$ of the hypercube $[-2^K, 2^K]^n$ and the triangulation $\mathcal{T}^{\rm std}$ outside of the hypercube $[-2^K, 2^K]^n$ with the mapping $\mathbf{x} \mapsto \rho\,\mathbf{x}$, where $\rho := 2^{-K}b$. We denote by $\mathcal{T}_{K,b}^{\rm std}$ the resulting set of n-simplices, i.e.
$$\mathcal{T}_{K,b}^{\rm std} = \rho\mathcal{T}_K^{\rm std} \cup \rho\left\{\mathfrak{S} \in \mathcal{T}^{\rm std} : \mathfrak{S} \cap] - 2^K, 2^K[^n = \emptyset\right\}.$$
4. As a final step define
$$\mathcal{T}_{K,b}^{\mathcal{C}} := \left\{\mathfrak{S}_\nu \in \mathcal{T}_{K,b}^{\rm std} \,:\, \mathfrak{S}_\nu \cap \mathcal{C}^\circ \neq \emptyset\right\}$$
   and set
$$\mathcal{D} := \bigcup_{\mathfrak{S}_\nu \in \mathcal{T}_{K,b}^{\mathcal{C}}} \mathfrak{S}_\nu.$$

Lemma 2. *Consider the sets $\mathcal{C}$ and $\mathcal{D}$ from the last definition. Then $\mathcal{D} \supset \mathcal{C}$ and $\mathcal{D} \in \mathcal{N}$.*

Proof: $\mathcal{D}$ is a closed set containing $\mathcal{C}^\circ$ and thus contains $\mathcal{C} \in \mathcal{N}$ because $\mathcal{C} = \overline{\mathcal{C}^\circ}$ by property iii) in Definition 1, so $\mathcal{C}$ is the smallest closed set containing $\mathcal{C}^\circ$. $\mathcal{D}$ fulfills properties i) and iii) of Definition 1 since $\mathcal{D}$ is a finite union of n-simplices. To see that property ii) of Definiton 1 is also fulfilled, i.e. that $\mathcal{D}^\circ$ is connected, notice the following: The definition of $\mathcal{T}_{K,b}^{\mathcal{C}}$ implies that for any $\mathfrak{S}_\nu \in \mathcal{T}_{K,b}^{\mathcal{C}}$ we have $\mathfrak{S}_\nu^\circ \cap \mathcal{C}^\circ \neq \emptyset$. Hence, any $\mathbf{x} \in \mathfrak{S}_\nu$ can be connected to a $\mathbf{y} \in \mathcal{C}^\circ$ with a line contained in $\mathfrak{S}_\nu^\circ$ with a possible exception of $\mathbf{x}$. Since $\mathcal{C} \in \mathcal{N}$, we have that $\mathcal{C}^\circ$ is connected. This implies that $\mathcal{D}^\circ$ is connected, too. □

Remark 14. For dimension $n = 2$ this construction is the same one as in [10] and for any dimension n the simplicial fan $\mathcal{T}_K^{\rm std}$ is the same one as in [11]. In [10], the simplicial complex $\mathcal{T}_{K,b}^{\rm std}$ for $n = 2$ is depicted and in [11], the simplicial fan $\mathcal{T}_K^{\rm std}$ is depicted for $n = 3$.

Remark 15. The triangulation $\mathcal{T}^{\rm std}$ is studied in more detail in Sections 4.1 and 4.2 in [24], but with slightly different notations. A sometimes more intuitive description of $\mathfrak{S}_{\mathbf{z},\mathcal{J},\sigma}$ is the simplex $\{\mathbf{x} \in \mathbb{R}^n \,:\, 1 \geq x_{\sigma(1)} \geq \ldots \geq x_{\sigma(n)} \geq 0\}$, which is translated by $\mathbf{x} \mapsto \mathbf{x} + \mathbf{z}$ and then a minus-sign is put in front of the i-th entry of the resulting vectors, whenever $i \in \mathcal{J}$.

Remark 16. $\mathcal{T}_{K,b}^{\mathcal{C}}$ is truly a triangulation, i.e. two different simplices in $\mathcal{T}_{K,b}^{\mathcal{C}}$ intersect in a common face or not at all, as shown in Lemma 3.

Lemma 3. *Consider the set of simplices $\mathcal{T}^{\mathcal{C}}_{K,b}$ from Definition 13 and let $\mathfrak{S}_1 = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and $\mathfrak{S}_2 = \text{co}\{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_n\}$ be two of its simplices. Then* {simplemma}

$$\mathfrak{S}_1 \cap \mathfrak{S}_2 = \mathfrak{S}_3 := \text{co}\{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_m\}, \tag{8}$$ {simpeq}

where $\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_m$ are the vertices that are common to $\mathfrak{S}_1$ and $\mathfrak{S}_2$, i.e. $\mathbf{z}_i = \mathbf{x}_{\alpha(i)} = \mathbf{y}_{\beta(i)}$ for $\alpha, \beta \in S_n$ and $i = 1, \ldots, m$.

Proof: We split the proof into four cases. For $\mathbf{0} \in \mathfrak{S}_1 \cap \mathfrak{S}_2$, equation (8) was proven in Lemma 2.7 in [11]. For $\mathbf{0} \notin \mathfrak{S}_1$ and $\mathbf{0} \notin \mathfrak{S}_2$, equation (8) follows directly by Theorem 4.11 in [24]. For $\mathbf{0} \in \mathfrak{S}_1$ and $\mathbf{0} \notin \mathfrak{S}_2$, equation (8) follows also from Theorem 4.11 in [24] because $\mathfrak{S}_1 \cap \mathfrak{S}_2 = \mathfrak{S}_1 \cap \mathfrak{S}_2 \cap \{\mathbf{x} : \|\mathbf{x}\|_\infty = b\}$ and the intersection $\mathfrak{S}_1 \cap \mathfrak{S}_2$ looks just like an intersection of two simplices in $\rho\mathcal{T}^{\text{std}}$. The case $\mathbf{0} \notin \mathfrak{S}_1$ and $\mathbf{0} \in \mathfrak{S}_2$ follows analogically. □

Now we define the algorithm to compute CPA Lyapunov functions for systems with an exponentially stable equilibrium at the origin.

Definition 17 (The Algorithm). Consider the system (1) and let $\mathcal{C} \in \mathcal{N}$. The procedure to search for a Lyapunov function for the system is as follows: {alg}

1. Set $K = 0$, $b = 1$, and let B be a constant such that
$$B \geq \max_{m,r,s=1,2,\ldots,n} \sup_{\mathbf{z}\in\mathcal{C}} \left| \frac{\partial^2 f_m}{\partial x_r \partial x_s}(\mathbf{z}) \right|.$$
2. Generate a linear programming problem as in Definition 6 using the triangulation $\mathcal{T}^{\mathcal{C}}_{K,b}$ and setting $B_\nu := B$ for all $\mathfrak{S}_\nu \in \mathcal{T}^{\mathcal{C}}_{K,b}$.
3. If the linear programming problem has a feasible solution, then we can compute a Lyapunov function $V : \mathcal{D} \to \mathbb{R}$, $\mathcal{D} := \bigcup_{\mathfrak{S}_\nu \in \mathcal{T}^{\mathcal{C}}_{K,b}} \mathfrak{S}_\nu$, for the system as shown in Theorem 1 and we are finished. If the linear programming problem from Step 2 does not have a feasible solution, then replace K by $K+1$ and b by $3/4 \cdot b$, i.e. $K \leftarrow K+1$ and $b \leftarrow 3/4 \cdot b$, and repeat Step 2.

Remark 18. The algorithm in Definition 17 searches for a feasible solution to the linear programming problems as in Definition 6, using the triangulations $\mathcal{T}^{\mathcal{C}}_{0,1}, \mathcal{T}^{\mathcal{C}}_{1,3/4}, \mathcal{T}^{\mathcal{C}}_{2,(3/4)^2}, \ldots$ By defining the sequence of triangulations, {triser}

$$\mathcal{T}_k := \mathcal{T}^{\mathcal{C}}_{k,\left(\frac{3}{4}\right)^k}, \quad \text{for } k \in \mathbb{N}_0, \tag{9}$$ {trisereq}

we can rephrase the algorithm: Search for a feasible solution to the linear programming problems defined as in Definition 6 using the elements of the sequence $(\mathcal{T}_k)_{k\in\mathbb{N}_0}$ in successive order. When a feasible solution is found, then use it to parameterize a CPA Lyapunov function. Further, it is advantageous to define $\mathcal{D}_k := \bigcup_{\mathfrak{S}_\nu \in \mathcal{T}_k} \mathfrak{S}_\nu$ for $k \in \mathbb{N}_0$.

Remark 19. If better estimates for the B_ν than the uniform bound B from Step 1 in the algorithm are available, then these can and should be used.

4. Main result

The next theorem, the main result of this paper, is valid for more general sequences $(\mathcal{T}_k)_{k\in\mathbb{N}}$ of triangulations, where $\mathcal{T}_{k+1}$ is constructed from $\mathcal{T}_k$ by scaling and tessellating its simplices, than for the sequence $(\mathcal{T}_k)_{k\in\mathbb{N}_0}$ from Remark 18. However, it is quite difficult to get hold of the exact conditions that must be fulfilled in a simple way so we restrict the theorem to this specific sequence. First, we state a fundamental lemma, the results of which are used in the proof of Theorem 5.

{Wlemma}

Lemma 4. *Consider the system (1) and assume that the origin is an exponentially stable equilibrium of the system. Let $\widetilde{\mathcal{C}} \in \mathcal{N}$ be a set contained in the origin's basin of attraction. Then there exists a Lyapunov function $W : \widetilde{\mathcal{C}} \to \mathbb{R}$ in the sense of Definition 2 that additionally satisfies the following conditions:*

a) W is continuous on $\widetilde{\mathcal{C}}$ and C^2 on $\widetilde{\mathcal{C}}^\circ \setminus \{\mathbf{0}\}$. Further, all second order derivatives of W can be extended continuously to the boundary $\partial\widetilde{\mathcal{C}}$ of $\widetilde{\mathcal{C}}$.

b) There is a constant $C_1^* < +\infty$ such that

$$\sup_{\mathbf{x}\in\widetilde{\mathcal{C}}\setminus\{\mathbf{0}\}} \|\nabla W(\mathbf{x})\|_2 \leq C_1^*. \tag{10}$$ {Cdef}

c) For all $k \in \mathbb{N}_0$ define $\varepsilon(k) := \frac{1}{2}\left(\frac{3}{4}\right)^k$. For all k so large that $\widetilde{\mathcal{C}} \setminus \mathcal{B}_{\varepsilon(k)} \neq \emptyset$ define

$$A_k := \max_{i,j=1,2,\ldots,n} \left\{ \left| \frac{\partial^2 W}{\partial x_i \partial x_j}(\mathbf{x}) \right| : \mathbf{x} \in \widetilde{\mathcal{C}} \setminus \mathcal{B}_{\varepsilon(k)} \right\}. \tag{11}$$ {Akdef}

Then there are constants $A > 0$ and $K^* \in \mathbb{N}$ such that

$$A_k \leq A \left(\frac{4}{3}\right)^k \quad \text{for all } k \geq K^*. \tag{12}$$ {Ak}

d)

$$W(\mathbf{x}) \geq \|\mathbf{x}\|_2, \quad D^+W(\mathbf{x}) \leq -\|\mathbf{x}\|_2, \quad \text{and} \quad D^+W(\mathbf{x}) = \nabla W(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) \tag{13}$$ {West}

for all $\mathbf{x} \in \widetilde{\mathcal{C}}$, all $\mathbf{x} \in \widetilde{\mathcal{C}}^\circ$ and all $\mathbf{x} \in \widetilde{\mathcal{C}}^\circ \setminus \{\mathbf{0}\}$ respectively.

e) Let $J := D\mathbf{f}(\mathbf{0})$ be the Jacobian of $\mathbf{f}$ at the origin, $I \in \mathbb{R}^{n\times n}$ be the identity matrix and $Q \in \mathbb{R}^{n\times n}$ be the unique symmetric positive definite matrix that fulfills the Lyapunov equation $J^TQ + QJ = -I$. Then there are constants $\delta > 0$ and $s > 0$ such that

$$W(\mathbf{x}) = s \cdot \sqrt{\mathbf{x}^TQ\mathbf{x}} \quad \text{for all } \mathbf{x} \in \mathcal{B}_\delta. \tag{14}$$ {locW}

Proof: The idea of how to construct the function W is as follows: Locally, near the origin, W is defined by (14), and away from the origin by $W(\mathbf{x}) = a\int_0^\infty \|S_t\mathbf{x}\|_2^2\,dt$, where $a > 0$ and $S_t\mathbf{x}$ denotes the solution to (1) with initial value $\mathbf{x}$ at time $t = 0$. In between, W is a smooth interpolation of these two functions. For $n = 2$ the details of this construction are worked out in the proof of Theorem 3.3 in [10] and the extension to a general $n \geq 2$ is straight forward so we skip it here. More exactly:

The statements a)-c) are proved in steps 1-5 of the proof of Theorem 3.3 in [10] for $n = 2$. The proof for general $n \geq 2$ is practically identical, just replace 2 by n where appropriate. For a proof of d) see Remark 4. The statement in e) follows from the definition of W_{loc} in Step 1 in the proof of Theorem 3.3 in [10], generalized to an arbitrary n, and d). □

Remark 20. The second order derivatives of W will in general diverge at the origin, but at a predictable rate as stated by (12).

Now we are ready for the main result of this paper.

{MAIN}

Theorem 5. *Consider the system (1) and assume that the origin is an exponentially stable equilibrium of the system. Let $\mathcal{C} \in \mathcal{N}$ be contained in the origin's basin of attraction. Then the algorithm from Definition 17 succeeds in computing a CPA Lyapunov function for the system in a finite number of steps.*

Proof: By Remark 18, it suffices to show that there is a $\mathfrak{K} \in \mathbb{N}$ such that any linear programming problem constructed as in Definition 6 using the triangulation $\mathcal{T}_k$ for any $k \geq \mathfrak{K}$ has a feasible solution. The set $\mathcal{C}$ in $\mathcal{T}_{K,b}^{\mathcal{C}}$ is the same $\mathcal{C}$ as in the statement of the theorem. Our proof further uses the Lyapunov function $W : \widetilde{\mathcal{C}} \to \mathbb{R}$ and C_1^*, K^*, $\varepsilon(k)$, A, and A_k from Lemma 4. The domain $\widetilde{\mathcal{C}}$ of W will be defined in Step 2 of the proof. We split the proof into eight steps.

1. **Definition of h_k**
   For every integer k define
   $$h_k := \frac{1}{2^k}\left(\frac{3}{4}\right)^k. \tag{15}$$ {defhk}
   Note that h_k is the scale down factor of the simplices in $\mathcal{T}_k$ not containing the origin. Further note that for every $\mathfrak{S}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{T}^k$, $\mathbf{0} \notin \mathfrak{S}_\nu$, we have
   $$\begin{aligned} &\min_{\substack{i,j=0,1,\ldots,n \\ i\neq j}} \|\mathbf{x}_i - \mathbf{x}_j\|_2 = h_k \quad \text{and} \\ &\max_{i,j=0,1,\ldots,n} \|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{n}\, h_k. \end{aligned} \tag{16}$$ {h_kmax}
2. **$\widetilde{\mathcal{C}}$ and $\mathcal{C} \subset \mathcal{D}_k \subset \widetilde{\mathcal{C}}$**
   Denote by $\mathcal{A}$ the basin of attraction of the equilibrium at the origin. Since $\mathcal{C} \subset \mathcal{A}$ is compact and $\mathcal{A}$ open, cf. e.g. Lemma 8.1 in [22], there is a positive Euclidean distance between $\mathcal{C}$ and the boundary of $\mathcal{A}$. Let $d > 0$ denote this distance if it is finite and otherwise set $d := 1$. Now define $\widetilde{\mathcal{C}}$ to be the set of all $\mathbf{x} \in \mathbb{R}^n$ that have Euclidean distance less than or equal to $d/2$ to $\mathcal{C}$. Clearly $\widetilde{\mathcal{C}} \in \mathcal{N}$. For all large enough $k \in \mathbb{N}_0$ the Euclidean distance from the boundary of $\mathcal{D}_k$, defined in Remark 18, to $\mathcal{C}$ is bounded by $\sqrt{n}\, h_k$, so there is a $K^{**} \in \mathbb{N}_0$ such that $\mathcal{C} \subset \mathcal{D}_k \subset \widetilde{\mathcal{C}}$ for every $k \geq K^{**}$.
3. **Estimate on $\|X_{k,\nu}^{-1}\|_1$ when $\mathbf{0} \notin \mathfrak{S}_{k,\nu}$**
   Choose an arbitrary n-simplex $\text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{T}^{\text{std}}$ and construct the matrix X by writing the components of the vector $\mathbf{x}_i - \mathbf{x}_0$ in its i-th row for $i = 1, 2, \ldots, n$. Note

that up to translations there are no more than $2^n \cdot n!$ different simplices in $\mathcal{T}^{\text{std}}$ and because there are $(n+1)!$ possibilities of ordering the vertices of any such simplex, there is only a finite number of possibilities of forming such a matrix X. Further, all of them are invertible. This means that we can define $\alpha > 0$ as the minimum eigenvalue of all possible $X^T X$ and then for any such X we have $\|X^{-1}\|_2 \le 1/\sqrt{\alpha}$. Define $X^* := \sqrt{n/\alpha}$. Now, for every $\mathfrak{S}_{k,\nu} \in \mathcal{T}_k$, $\mathbf{0} \notin \mathfrak{S}_{k,\nu}$, we can construct a matrix $X_{k,\nu}$ as in Remark 9 and $h_k^{-1}\mathfrak{S}_{k,\nu}$ is, up to a possible translation, equal to a simplex in $\mathcal{T}^{\text{std}}$. Thus, the matrix $X_{k,\nu}/h_k$ corresponds to a matrix X as above and therefore $\|X_{k,\nu}^{-1}\|_2 \le 1/(h_k\sqrt{\alpha})$. Hence,

$$\|X_{k,\nu}^{-1}\|_1 \le \sqrt{n}\,\|X_{k,\nu}^{-1}\|_2 \le \frac{X^*}{h_k} \tag{17}$$ {Xm1est}

for all k and ν. Note especially that X^* is a constant independent of k and ν.

4. **Estimate on $\|X_{k,\nu}^{-1}\mathbf{w}_{k,\nu}^* - \nabla W(\mathbf{x}_i)\|_1$ when $\mathbf{0} \notin \mathfrak{S}_{k,\nu}$**
Let $k \ge \max\{K^*, K^{**}\}$ and $\mathfrak{S}_{k,\nu} = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{T}_k$ and define

$$\mathbf{w}_{k,\nu}^* := \begin{pmatrix} W(\mathbf{x}_1) - W(\mathbf{x}_0) \\ W(\mathbf{x}_2) - W(\mathbf{x}_0) \\ \vdots \\ W(\mathbf{x}_n) - W(\mathbf{x}_0) \end{pmatrix}. \tag{18}$$ {wknu}

We will need upper bounds on $\|X_{k,\nu}^{-1}\mathbf{w}_{k,\nu}^* - \nabla W(\mathbf{x}_i)\|_1$ later on, for $i = 0, 1, 2, \ldots, n$ if $\mathbf{0} \notin \mathfrak{S}_{k,\nu}$ and for $i = 1, 2, \ldots, n$ if $\mathbf{0} \in \mathfrak{S}_{k,\nu}$. Here, we derive the appropriate bounds if $\mathbf{0} \notin \mathfrak{S}_{k,\nu}$ and in Step 5 we consider the case $\mathbf{0} \in \mathfrak{S}_{k,\nu}$, which is quite different.
Assume $\mathbf{0} \notin \mathfrak{S}_{k,\nu}$. Note that in this case $\mathfrak{S}_{k,\nu} \subset \widetilde{\mathcal{C}} \setminus \mathcal{B}_{\varepsilon(k)}$ by the definition of $\varepsilon(k)$ and the construction of the triangulation $\mathcal{T}_k$. Moreover, W is two times continuously differentiable on $\mathfrak{S}_{k,\nu}$ and for $i = 1, 2, \ldots, n$ we have by Taylor's theorem

$$W(\mathbf{x}_i) = W(\mathbf{x}_0) + \nabla W(\mathbf{x}_0) \cdot (\mathbf{x}_i - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x}_i - \mathbf{x}_0)^T H_W(\mathbf{z}_i)(\mathbf{x}_i - \mathbf{x}_0),$$

where H_W is the Hessian of W and $\mathbf{z}_i = \mathbf{x}_0 + \vartheta_i(\mathbf{x}_i - \mathbf{x}_0)$ for some $\vartheta_i \in]0,1[$.
By rearranging terms and combining this delivers

$$\mathbf{w}_{k,\nu}^* - X_{k,\nu}\nabla W(\mathbf{x}_0) = \frac{1}{2}\begin{pmatrix} (\mathbf{x}_1 - \mathbf{x}_0)^T H_W(\mathbf{z}_1)(\mathbf{x}_1 - \mathbf{x}_0) \\ (\mathbf{x}_2 - \mathbf{x}_0)^T H_W(\mathbf{z}_2)(\mathbf{x}_2 - \mathbf{x}_0) \\ \vdots \\ (\mathbf{x}_n - \mathbf{x}_0)^T H_W(\mathbf{z}_n)(\mathbf{x}_n - \mathbf{x}_0) \end{pmatrix}. \tag{19}$$ {wmXnW}

With $H_W(\mathbf{z}) = (h_{ij}(\mathbf{z}))_{i,j=1,2,\ldots,n}$ we have that $\max_{\mathbf{z}\in\mathfrak{S}_{k,\nu}} |h_{ij}(\mathbf{z})| \le A_k$ by (11) because $\mathfrak{S}_{k,\nu} \subset \widetilde{\mathcal{C}} \setminus \mathcal{B}_{\varepsilon(k)}$. Hence, e.g. by Lemma 4.2 [1], we have

$$\max_{\mathbf{z}\in\mathfrak{S}_{k,\nu}} \|H_W(\mathbf{z})\|_2 \quad \le \quad nA_k. \tag{20}$$ {AA}

By (16), (12), and (15), we obtain

$$\left|(\mathbf{x}_i - \mathbf{x}_0)^T H_W(\mathbf{z}_i)(\mathbf{x}_i - \mathbf{x}_0)\right| \le (\sqrt{n}h_k)^2 \|H_W(\mathbf{z}_i)\|_2 \le n^2 A_k h_k^2 \le n^2 \frac{A}{2^k} h_k.$$

Hence, by (19),

$$\|\mathbf{w}^*_{k,\nu} - X_{k,\nu}\nabla W(\mathbf{x}_0)\|_1 \le \frac{n^3 A}{2^{k+1}} h_k. \qquad (21) \quad \{\texttt{Ain1}\}$$

Further, for $i, j = 1, 2, \ldots, n$ there is a $\mathbf{z}_{ij}$ on the line segment between $\mathbf{x}_i$ and $\mathbf{x}_0$, such that

$$\partial_j W(\mathbf{x}_i) - \partial_j W(\mathbf{x}_0) = \nabla \partial_j W(\mathbf{z}_{ij}) \cdot (\mathbf{x}_i - \mathbf{x}_0),$$

where $\partial_j W$ denotes the j-th component of ∇W and $\nabla \partial_j W$ is the gradient of this function. Then, by the definition of A_k we have

$$|\partial_j W(\mathbf{x}_i) - \partial_j W(\mathbf{x}_0)| \le \|\nabla \partial_j W(\mathbf{z}_{ij})\|_2 \|\mathbf{x}_i - \mathbf{x}_0\|_2 \le \sqrt{n} A_k \sqrt{n} h_k = n A_k h_k$$

so we have

$$\|\nabla W(\mathbf{x}_i) - \nabla W(\mathbf{x}_0)\|_1 \le n \cdot n A_k h_k \le \frac{n^2 A}{2^k}.$$

From this, (17), and (21) we obtain for $i = 0, 1, 2, \ldots, n$ the inequality

$$\begin{aligned} &\|X^{-1}_{k,\nu}\mathbf{w}^*_{k,\nu} - \nabla W(\mathbf{x}_i)\|_1 \\ \le\ & \|X^{-1}_{k,\nu}\mathbf{w}^*_{k,\nu} - \nabla W(\mathbf{x}_0)\|_1 + \|\nabla W(\mathbf{x}_i) - \nabla W(\mathbf{x}_0)\|_1 \\ \le\ & \|X^{-1}_{k,\nu}\|_1 \|\mathbf{w}^*_{k,\nu} - X_{k,\nu}\nabla W(\mathbf{x}_0)\|_1 + \frac{n^2 A}{2^k} \\ \le\ & \frac{X^*}{h_k}\frac{n^3 A}{2^{k+1}} h_k + \frac{n^2 A}{2^k} = \frac{n^2 A}{2^{k+1}}(nX^* + 2). \end{aligned} \qquad (22) \quad \{\texttt{xfirst}\}$$

A further useful consequence, which we need later, is that

$$\|X^{-1}_{k,\nu}\mathbf{w}^*_{k,\nu}\|_1 \le \|\nabla W(\mathbf{x}_i)\|_1 + \frac{n^2 A}{2^{k+1}}(nX^* + 2) \le \sqrt{n}\, C_1^* + \frac{n^2 A}{2^{k+1}}(nX^* + 2) \qquad (23) \quad \{\texttt{cfirst}\}$$

holds, where we used the bounds (10) on $\|\nabla W(\mathbf{x}_i)\|_2$.

5. **Estimate on $\|X^{-1}_{k,\nu}\mathbf{w}^*_{k,\nu} - \nabla W(\mathbf{x}_i)\|_1$ when $\mathbf{0} \in \mathfrak{S}_{k,\nu}$**
We will show that there is a constant $C_2^* > 0$ and a $K^{***} \in \mathbb{N}_0$, such that

$$\|X^{-1}_{k,\nu}\mathbf{w}^*_{k,\nu} - \nabla W(\mathbf{x}_i)\|_1 \le C_2^* 2^{-k} \quad \text{for all } k \ge K^{***} \text{ and all } \mathfrak{S}_\nu \in \mathcal{T}_k. \qquad (24) \quad \{\texttt{bekannt}\}$$

This estimate follows from the inequality at the end of Step 7 in the proof of Theorem 3.2 in [11] and Lemma 4 e). We explain this in more detail.
In Step 7 in the proof of Theorem 3.2 in [11] it is shown that

$$\|\nabla w(\mathbf{x}_i) - \nabla v(\mathbf{x}_i)\|_2 \le \frac{\rho}{b} C_0, \qquad (25) \quad \{\texttt{oldest}\}$$

where C_0 is a constant. Further, $\rho/b = 2^{-K}$ and the estimate (25) holds true for all $b > 0$ small enough and $K \in \mathbb{N}$ large enough. Note that in the proof of Theorem 3.2 in [11], the values of b and K are fixed in Step 2 and Step 4, respectively. Let us consider such a fixed pair $\widetilde{b}$ and $\widetilde{K}$. Following the proof in [11], it is easy to see that the estimate (25) holds true for all b such that $0 < b < \widetilde{b}$ and all $K \in \mathbb{N}_0$ with $K \ge \widetilde{K}$.

In our notation, (25) reads

$$\left\| \frac{1}{s} X_{k,\nu}^{-1} \mathbf{w}_{k,\nu}^* - \frac{1}{s} \nabla W(\mathbf{x}_i) \right\|_2 \leq 2^{-K} C_0. \tag{26}$$ {oldest2}

since $\nabla w(\mathbf{x}_i)$ in (25) corresponds to $\frac{1}{s} X_{k,\nu}^{-1} \mathbf{w}_{k,\nu}^*$ in our notation, see Remark 9, and $\nabla v(\mathbf{x}_i)$ in (25) corresponds to $\frac{1}{s} \nabla W(\mathbf{x}_i)$ in our notation. Here, $s > 0$ is the constant from Lemma 4 e).

Consider the sequence of triangulations $(\mathcal{T}_k)_{k \in \mathbb{N}_0}$ from Remark 18 and recall that the simplicial fan of $\mathcal{T}_k$ triangulates the cube $[-(3/4)^k, (3/4)^k]^n$. Now, for all $k \in \mathbb{N}_0$ such that

$$[-(3/4)^k, (3/4)^k]^n \subset \mathcal{B}_\delta, \quad (3/4)^k \leq \widetilde{b}, \quad \text{and} \quad k \geq \widetilde{K}, \tag{27}$$ {kss}

the estimate (26) holds true with $b := (3/4)^k$ and $K := k$. Here $\delta > 0$ is the constant from Lemma 4 e), and $X_{k,\nu}$ and $\mathbf{w}_{k,\nu}^*$ are constructed for an arbitrary $\mathfrak{S}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{T}_k$ as in Remark 9 and (18), respectively. Hence, there is a $K^{***} \in \mathbb{N}_0$ and a constant C_2^* such that the assertion (24) holds true, e.g. with K^{***} as the smallest k such that (27) holds true and $C_2^* = \sqrt{n}\, s\, C_0$.

Similar to (23) we can deduce from (24) that

$$\|X_{k,\nu}^{-1} \mathbf{w}_{k,\nu}^*\|_1 \leq \|\nabla W(\mathbf{x}_i)\|_1 + C_2^* 2^{-k} \leq \sqrt{n}\, C_1^* + C_2^* 2^{-k}. \tag{28}$$ {cfirst2}

6. **Assign values to the linear program**
In this step we assign values to the variables of the linear programming problem from Definition 6 used by the algorithm in Definition 17. We then show that the constraints (4), (5), and (6) are fulfilled if the simplices in the triangulation $\mathcal{T}_k$ are small enough, i.e. for all large enough $k \in \mathbb{N}_0$. To do this, let $k \geq \max\{K^*, K^{**}, K^{***}\}$ be arbitrary but fixed.
The assignments are as follows. For every $\mathfrak{S}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{T}_k$ we set:
   i) $V_{\mathbf{x}_i} := 2W(\mathbf{x}_i)$ for every vertex $\mathbf{x}_i$ of $\mathfrak{S}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where W is the Lyapunov function from Lemma 4.
   ii) $C_{\nu,i} := \left|\mathbf{e}_i^T X_\nu^{-1} \mathbf{v}_\nu^*\right|$ for $i = 1, 2, \ldots, n$, where X_ν and $\mathbf{v}_\nu^*$ are constructed for $\mathfrak{S}_\nu$ and $V_{\mathbf{x}_i}$ as in Remark 9.

   By doing this, we have assigned values to all the variables of the linear programming problem. The factor 2 in i) is necessary because we need V not merely to fulfill $-\|\mathbf{x}_i\|_2 \geq \nabla V_\nu \cdot \mathbf{f}(\mathbf{x}_i)$, but the stronger inequality (6).
   Clearly $V_{\mathbf{0}} = 2W(\mathbf{0}) = 0$ and $V_{\mathbf{x}_i} = 2W(\mathbf{x}_i) \geq 2\|\mathbf{x}_i\|_2 \geq \|\mathbf{x}_i\|_2$ by (13) for every $\mathfrak{S}_\nu \in \mathcal{T}_k$ and every vertex $\mathbf{x}_i$ of $\mathfrak{S}_\nu$. Therefore, the constraints (4) are fulfilled. Further, by Remark 9, $\nabla V_{\nu,i} = \mathbf{e}_i^T X_\nu^{-1} \mathbf{v}_\nu^*$, where $\nabla V_{\nu,i}$ is the i-th component of ∇V_ν, so $C_{\nu,i} = \left|\mathbf{e}_i^T X_\nu^{-1} \mathbf{v}_\nu^*\right| \geq |\nabla V_{\nu,i}|$ holds trivially. Hence, the constraints (5) are fulfilled. The challenge is to show that the constraints (6) are fulfilled. We show this for an arbitrary simplex $\mathfrak{S}_\nu$, distinguishing between the two cases that $\mathfrak{S}_\nu$ is not in the simplicial fan of $\mathcal{T}_k$ (Step 7) and that $\mathfrak{S}_\nu$ is in the simplicial fan (Step 8).

7. **Constraints (6) when $\mathbf{0} \notin \mathfrak{S}_\nu$**
Pick an arbitrary $\mathfrak{S}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{T}_k$ such that $\mathbf{0} \notin \mathfrak{S}_\nu$. We notice that

$$\nabla V_\nu = X_\nu^{-1} \mathbf{v}_\nu^* = 2X_{k,\nu}^{-1} \mathbf{w}_{k,\nu}^*,$$

where $\mathbf{w}^*_{k,\nu}$ is defined as in (18) and we emphasize the k-dependence of $X_\nu = X_{k,\nu}$ as in Step 4. Let $L > 0$ be a Lipschitz constant for $\mathbf{f}$ on $\widetilde{\mathcal{C}}$ such that $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\|_\infty \leq L\|\mathbf{x} - \mathbf{y}\|_2$ for all $\mathbf{x}, \mathbf{y} \in \widetilde{\mathcal{C}}$.
By (13) and the estimate (22) we have for $i = 0, 1, \dots, n$ that

$$\begin{aligned}\nabla V_\nu \cdot \mathbf{f}(\mathbf{x}_i) &= 2X^{-1}_{k,\nu}\mathbf{w}^*_{k,\nu} \cdot \mathbf{f}(\mathbf{x}_i)\\ &= 2\nabla W(\mathbf{x}_i) \cdot \mathbf{f}(\mathbf{x}_i) + 2\left(X^{-1}_{k,\nu}\mathbf{w}^*_{k,\nu} - \nabla W(\mathbf{x}_i)\right) \cdot \mathbf{f}(\mathbf{x}_i)\\ &\leq -2\|\mathbf{x}_i\|_2 + 2\|X^{-1}_{k,\nu}\mathbf{w}^*_{k,\nu} - \nabla W(\mathbf{x}_i)\|_1 \|\mathbf{f}(\mathbf{x}_i)\|_\infty\\ &\leq -2\|\mathbf{x}_i\|_2 + \frac{n^2 A}{2^k}(nX^* + 2) \cdot L\|\mathbf{x}_i\|_2.\end{aligned}$$

Hence, the constraints (6), i.e.

$$-\|\mathbf{x}_i\|_2 \geq \nabla V_\nu \cdot \mathbf{f}(\mathbf{x}_i) + E_{\nu,i}\sum_{j=1}^{n} C_{\nu,j}$$

are fulfilled whenever k is so large that

$$-\|\mathbf{x}_i\|_2 \geq -2\|\mathbf{x}_i\|_2 + \frac{n^2 AL}{2^k}(nX^* + 2) \cdot \|\mathbf{x}_i\|_2 + E_{\nu,i}\sum_{j=1}^{n} C_{\nu,j},$$

which is equivalent to

$$1 \geq \frac{n^2 AL}{2^k}(nX^* + 2) + \frac{1}{\|\mathbf{x}_i\|_2} E_{\nu,i}\sum_{j=1}^{n} C_{\nu,j}. \tag{29}$$ {uff1}

Because $\mathbf{0} \notin \mathfrak{S}_\nu$ we have by (7) and (16) that

$$\begin{aligned}E_{\nu,i} &:= \|\mathbf{x}_i - \mathbf{x}_0\|_2 \left(\max_{j=1,2,\dots,n} \|\mathbf{x}_j - \mathbf{x}_i\|_2 + \|\mathbf{x}_i - \mathbf{x}_0\|_2\right)\frac{nB_\nu}{2}\\ &\leq \sqrt{n}\, h_k(2\sqrt{n}\, h_k) \cdot \frac{nB_\nu}{2} = n^2 h_k^2 B_\nu\end{aligned}$$

and $\|\mathbf{x}_i\|_2 \geq (3/4)^k$ because $\mathfrak{S}_\nu$ is not in the simplicial fan of $\mathcal{T}_k$. Further, by (23)

$$\sum_{j=1}^{n} C_{\nu,j} = 2\|X^{-1}_{k,\nu}\mathbf{w}^*_{k,\nu}\|_1 \leq 2\sqrt{n}\, C_1^* + \frac{n^2 A}{2^k}(nX^* + 2). \tag{30}$$ {lastC}

Thus, (29) holds true if,

$$\begin{aligned}1 &\geq \frac{n^2 AL}{2^k}(nX^* + 2) + \left(\frac{4}{3}\right)^k \cdot n^2 h_k^2 B_\nu \cdot \left(2\sqrt{n}\, C_1^* + \frac{n^2 A}{2^k}(nX^* + 2)\right)\\ &= \frac{n^2 AL}{2^k}(nX^* + 2) + n^2 B_\nu \left(2\sqrt{n}\, C_1^* + \frac{n^2 A}{2^k}(nX^* + 2)\right)\frac{1}{2^{2k}}\left(\frac{3}{4}\right)^k,\end{aligned}$$

where we have used the formula (15) for h_k. This last inequality clearly holds true for all large enough k.

8. **Constraints (6) when $\mathbf{0} \in \mathfrak{S}_\nu$**
We now consider an arbitrary $\mathfrak{S}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{T}_k$, with $\mathbf{x}_0 = \mathbf{0}$. Define $\mathbf{w}^*_{k,\nu}$ and $X_{k,\nu}$ exactly as in the last step.
By (7),

$$E_{\nu,0} := \frac{nB_\nu}{2}\|\mathbf{x}_0 - \mathbf{x}_0\|_2 \left(\max_{j=1,2,\ldots,n} \|\mathbf{x}_j - \mathbf{x}_0\|_2 + \|\mathbf{x}_0 - \mathbf{x}_0\|_2\right) = 0$$

so the linear constraints (6) are automatically fulfilled for $i = 0$, because the condition is

$$-\underbrace{\|\mathbf{0}\|_2}_{=0} \geq \nabla V_\nu \cdot \underbrace{\mathbf{f}(\mathbf{0})}_{=0} + \underbrace{E_{\nu,0}}_{=0} \sum_{j=1}^n C_{\nu,j},$$

i.e. $0 \geq 0$.
For $i = 1, 2, \ldots, n$, the constraints (6) read

$$-\|\mathbf{x}_i\|_2 \geq \nabla V_\nu \cdot \mathbf{f}(\mathbf{x}_i) + E_{\nu,i} \sum_{j=1}^n C_{\nu,j}.$$

Similar to Step 7, but using the estimate (24) instead of (22), we get

$$\begin{aligned}\nabla V_\nu \cdot \mathbf{f}(\mathbf{x}_i) &= 2X^{-1}_{k,\nu}\mathbf{w}^*_{k,\nu} \cdot \mathbf{f}(\mathbf{x}_i) \\ &= 2\nabla W(\mathbf{x}_i) \cdot \mathbf{f}(\mathbf{x}_i) + 2\left(X^{-1}_{k,\nu}\mathbf{w}^*_{k,\nu} - \nabla W(\mathbf{x}_i)\right) \cdot \mathbf{f}(\mathbf{x}_i) \\ &\leq -2\|\mathbf{x}_i\|_2 + 2\|X^{-1}_{k,\nu}\mathbf{w}^*_{k,\nu} - \nabla W(\mathbf{x}_i)\|_1 \cdot \|\mathbf{f}(\mathbf{x}_i)\|_\infty \\ &\leq -2\|\mathbf{x}_i\|_2 + 2C_2^* 2^{-k} \cdot L\|\mathbf{x}_i\|_2.\end{aligned}$$

Thus, the constraints are fulfilled if

$$-\|\mathbf{x}_i\|_2 \geq -2\|\mathbf{x}_i\|_2 + 2C_2^* 2^{-k} \cdot L\|\mathbf{x}_i\|_2 + E_{\nu,i} \sum_{j=1}^n C_{\nu,j},$$

which is equivalent to

$$1 \geq 2C_2^* L 2^{-k} + \frac{1}{\|\mathbf{x}_i\|_2} E_{\nu,i} \sum_{j=1}^n C_{\nu,j}. \qquad (31) \quad \{\texttt{uff2}\}$$

Since $\mathbf{x}_0 = \mathbf{0}$ and $\|\mathbf{x}_j\|_\infty = (3/4)^k$ for $j = 1, 2, \ldots, n$, we now have

$$E_{\nu,i} := \frac{nB_\nu}{2}\|\mathbf{x}_i - \mathbf{x}_0\|_2 \left(\max_{j=1,2,\ldots,n} \|\mathbf{x}_j - \mathbf{x}_0\|_2 + \|\mathbf{x}_i - \mathbf{x}_0\|_2\right) \leq n^{\frac{3}{2}} B_\nu \left(\frac{3}{4}\right)^k \|\mathbf{x}_i\|_2.$$

Finally, notice that similar to (30) we get by (28) that

$$\sum_{j=1}^n C_{\nu,j} = 2\|X^{-1}_{k,\nu}\mathbf{w}^*_{k,\nu}\|_1 \leq 2\sqrt{n}\, C_1^* + \frac{2C_2^*}{2^k}$$

so (31) is fulfilled if

$$1 \geq 2C_2^* L 2^{-k} + n^{\frac{3}{2}} B \left(\frac{3}{4}\right)^k \left(2\sqrt{n}\, C_1^* + \frac{2C_2^*}{2^k}\right).$$

Again this inequality clearly holds true for all large enough k.

The conclusions are: In Step 6 we assigned values to the variables of the linear programming problem from Definition 6 and showed that for all $k \geq \max\{K^*, K^{**}, K^{***}\}$ the linear constraints (4) and (5) of the linear programming problem are fulfilled. In Steps 7 and 8 we proved that for all large enough $k \geq \max\{K^*, K^{**}, K^{***}\}$ the constraints (6) of the linear programming problem are fulfilled. Hence, there is a $\mathfrak{K} \geq \max\{K^*, K^{**}, K^{***}\}$ such that the linear programming problem has a feasible solution using the triangulation $\mathcal{T}_k$ whenever $k \geq \mathfrak{K}$. By Theorem 1 we can use such a feasible solution to parameterize a CPA Lyapunov function for the system (1). Since the algorithm from Definition 17 tries to find a feasible solution to the linear programming problem from Definition 6 using the triangulations $\mathcal{T}_k$ for $k = 0, 1, 2, \ldots$, and there are algorithms, e.g. the Simplex algorithm, that always find a solution to a linear programming problem whenever it possesses a feasible solution, we have shown that the algorithm from Definition 17 always succeeds in parameterizing a CPA Lyapunov function for the system (1). □

5. Examples

{examples}

5.1. Van der Pol

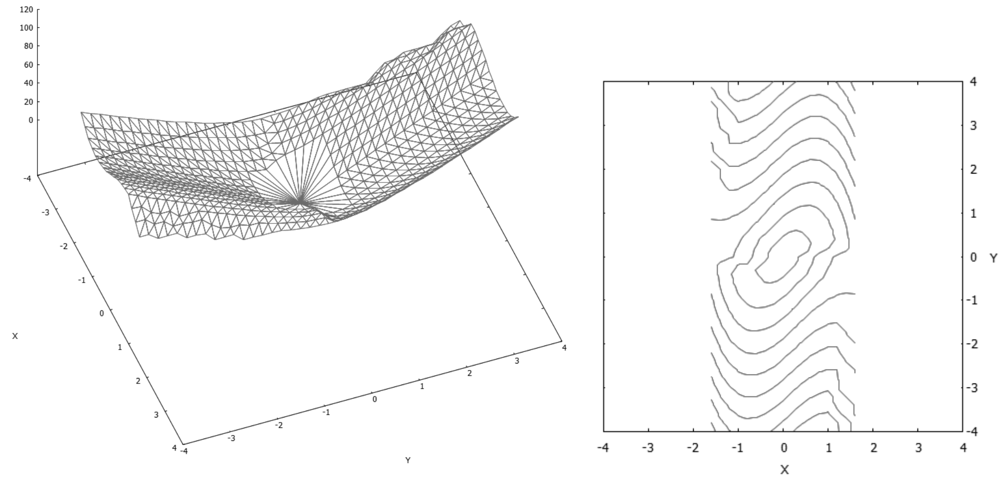
Consider the time-reversed van der Pol equation $\dot{x} = -y$, $\dot{y} = x + (1 - x^2)y$ from [2], i.e. the system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ with

$$\mathbf{f}(x_1, x_2) = \begin{pmatrix} -x_2 \\ x_1 + (1 - x_1^2)x_2 \end{pmatrix}.$$

We computed a CPA Lyapunov function $V : [-4, 4] \times [-1.6, 1.6] \to \mathbb{R}$ for this system. The parameters used for the triangulation $\mathcal{T}_{K,b}^{\mathcal{C}}$ in Definition 13 were $\mathcal{C} = [-4, 4] \times [-1.6, 1.6]$, $b = 0.8$, and $K = 2$. We used this triangulation in the linear programming problem in Definition 6 and because

$$\max_{m,r,s=1,2} \max_{\mathbf{x} \in \mathfrak{S}_\nu} \left| \frac{\partial^2 f_m}{\partial x_r \partial x_s}(\mathbf{z}) \right| = \max_{\mathbf{x} \in \mathfrak{S}_\nu} \max\{|-2x_2|, |-2x_1|\}$$

we can set $B_\nu := 2\max_{\mathbf{x} \in \mathfrak{S}_\nu} \max\{|x_1|, |x_2|\}$ for all $\mathfrak{S}_\nu \in \mathcal{T}$, cf. Step 4 in (6). Here it is very easy to derive exact tight bounds on the second-order derivatives of $\mathbf{f}$, but even if it is not, it is usually not difficult to deliver some upper bounds. If bounds are more generous, one might need a finer triangulation. For the above parameters, the linear programming problem has a feasible solution. A computed CPA Lyapunov and some of its level sets are depicted in Figure 1.



(a) CPA Lyapunov function. (b) Some of its level sets.

Figure 1: Computed CPA Lyapunov function and some of its level sets for a time-reversed van der Pol equation. Each of the closed level sets is the boundary of a sublevel set which is a subset of the basin of attraction of the origin. The computation takes less than 13 sec. on a state of the art PC. {fig1}

5.2. 3-dimensional Example

As a second example we consider the system $\dot{x} = -x - y - z$, $\dot{y} = \sin x - 2y(1+x) + z$, $\dot{z} = x(1+x) + y - 2\sin z$, i.e. $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ with

$$\mathbf{f}(x_1, x_2, x_3) = \begin{pmatrix} -x_1 - x_2 - x_3 \\ \sin(x_1) - 2x_2(1 + x_1) + x_3 \\ x_1(1 + x_1) + x_2 - 2\sin(x_3) \end{pmatrix}.$$

We computed a CPA Lyapunov function $V : [-0.5, 0.5]^3 \to \mathbb{R}$ for this system. The parameters used for the triangulation $\mathcal{T}^{\mathcal{C}}_{K,b}$ in Definition 13 were $\mathcal{C} = [-0.5, 0.5]^3$, $b = 0.5$, and $K = 0$. Thus $\mathcal{T}^{\mathcal{C}}_{K,b} = 0.5\mathcal{T}^{std}$ in $\mathcal{C}$. We used this triangulation in the linear programming problem in Definition 6 and because

$$\max_{m,r,s=1,2,3} \max_{\mathbf{x}\in\mathfrak{S}_\nu} \left| \frac{\partial^2 f_m}{\partial x_r \partial x_s}(\mathbf{z}) \right| = \max_{\mathbf{x}\in\mathfrak{S}_\nu} \max\{|-\sin(x_1)|, 2, |-2|, |2\sin(x_3)|\}$$

we can set $B_\nu := 2$ for all $\mathfrak{S}_\nu \in \mathcal{T}$, cf. Step 4 in (6). For these parameters the linear programming problem has a feasible solution. A level set of the computed CPA Lyapunov function is depicted in Figure 2.

6. Summary and further work {summary}

With this paper the authors close the theoretical and algorithmical part of a project started in 2009: We have now revised the CPA method to compute Lyapunov functions for systems with exponentially stable equilibria to include the equilibrium in the domain of

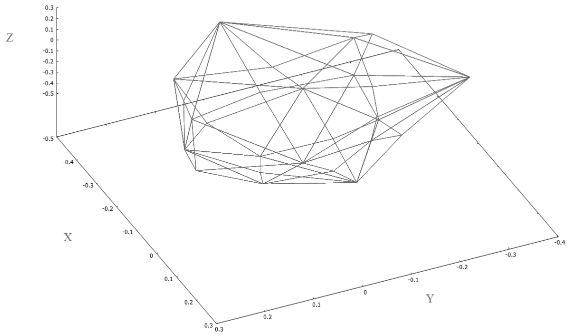


Figure 2: A level set of a CPA Lyapunov function for the system $\dot{x} = -x - y - z$, $\dot{y} = \sin x - 2y(1 + x) + z$, $\dot{z} = x(1+x)+y-2\sin z$. The sublevel set is a subset of the basin of attraction of the origin. The computation takes less than a second on a state of the art PC. {fig2}

the CPA Lyapunov function. Designing such a revised algorithm turned out to be a much harder problem than anticipated, but the results are very satisfactory and not only solve the problem aimed at but give a much deeper understanding of the CPA method itself. For the full solution, as presented in this paper, a series of partially very technical details had to be worked out. Former publications by the authors, [9, 10, 11], in addition to presenting important results in themselves, worked out many of these details. In particular, the revised CPA algorithm for planar systems was presented in [10]. In Step 9 in the proof of the main theorem of [10] we had to rely explicitly on the system being planar and the extension to higher dimensions needed a completely different methodology.

We have implemented the revised CPA method for n-dimensional systems and have applied it to examples. In Figure 1, a planar example of a computed CPA Lyapunov function and some of its level sets are presented, and in Figure 2 a level set of a computed CPA Lyapunov function for a three-dimensional system is depicted. Our program is written in C++ and makes use of the Armadillo linear algebra library [34] and the GNU Linear Programming Kit (http://www.gnu.org/software/glpk/glpk.html). Our software is in the stage of being properly debugged, tested and documented, and will be made available free of charge on the internet in the near future. The goal is to be able to equip the scientific community with an easy-to-use tool to compute CPA Lyapunov functions for general nonlinear systems with exponentially stable equilibria.

[1] R. Baier, L. Grüne and S. Hafstein, Linear programming based Lyapunov function computation for differential inclusions, Discrete Contin. Dyn. Syst. Ser. B, **17** (2012), 33–56.

[2] H. Ban and W. Kalies, A computational approach to Conley's decomposition theorem, J. Comput. Nonlinear Dynam., **1-4** (2006), 312–319.

[3] R. Bartels and G. Stewart, Algorithm 432: Solution of the matrix equation $AX+XB = C$, Comm. ACM, **15** (1972), 820–826.

[4] D. Bertsimas and J. Tsitsiklis, "Introduction to linear optimization", Athena Scientific, 1997.

[5] F. Clarke, Y. Ledyaev and R. Stern, Asymptotic stability and smooth Lyapunov functions, J. Differential Equations, **149** (1998), 69–114.

[6] C. Conley "Isolated invariant sets and the Morse index", Vol. 38 of CBMS, American Math. Soc., 1978.

[7] N. Eghbal, N. Pariz and A. Karimpour, Discontinuous piecewise quadratic Lyapunov functions for planar piecewise affine systems, J. Math. Anal. Appl., **399** (2012), 586–593.

[8] P. Giesl, "Construction of Global Lyapunov Functions Using Radial Basis Functions", Lecture Notes in Mathematics **1904**, Springer, 2007.

[9] P. Giesl and S. Hafstein, Existence of piecewise affine Lyapunov functions in two dimensions. J. Math. Anal. Appl., **371** (2010), 233–248.

[10] P. Giesl and S. Hafstein, Construction of Lyapunov functions for nonlinear planar systems by linear programming, J. Math. Anal. Appl., **388** (2012), 463–479.

[11] P. Giesl and S. Hafstein, Existence of piecewise linear Lyapunov functions in arbitary dimensions, Discrete Contin. Dyn. Syst., **32** (2012), 3539–3565.

[12] L. Grüne, "Asymptotic Behavior of Dynamical and Control Systems under Perturbation and Discretization ", in: Lecture Notes in Mathematics **1783**, Springer, 2002.

[13] S. Hafstein, A constructive converse Lyapunov theorem on exponential stability. Discrete Contin. Dyn. Syst., **10** (2004), 657–678.

[14] S. Hafstein, A constructive converse Lyapunov theorem on asymptotic stability for nonlinear autonomous ordinary differential equations. Dynamical Systems, **20** (2005), 281–299

[15] S. Hafstein, "An algorithm for constructing Lyapunov functions", Electron. J. Differential Equ. Monogr., **8** (2007).

[16] W. Hahn, "Stability of Motion", Springer, 1967.

[17] M. Johansson and A. Rantzer, On the computation of piecewise quadratic Lyapunov functions, In: Proceedings of the 36th IEEE Conference on Decision and Control (1997)

[18] T. Johansen, Computation of Lyapunov Functions for Smooth Nonlinear Systems using Convex Optimization, Automatica, **36** (2000), 1617–1626.

[19] P. Julian, "A high level canonical piecewise linear representation: Theory and applications", Ph.D. Thesis: Universidad Nacional del Sur, Bahia Blanca, Argentina, 1999.

[20] P. Julian, J. Guivant and A. Desages, A parametrization of piecewise linear Lyapunov function via linear programming, Int. Journal of Control, **72** (1999), 702–715.

[21] W. Kalies, K. Mischaikow and R. VanderVorst, An algorithmic approach to chain recurrence, Found. Comput. Math., **5-4** (2005), 409–449.

[22] H. Khalil, "Nonlinear Systems", Prentice Hall, 3. ed., 2002.

[23] A. Lyapunov, "The general problem of the stability of motion", translated by A. T. Fuller, Taylor & Francis, 1992.

[24] S. Marinosson, "Stability analysis of nonlinear systems with linear programming: A Lyapunov functions based approach", Ph.D. Thesis: Gerhard-Mercator-University, Duisburg, Germany, 2002.

[25] S. Marinosson, Lyapunov function construction for ordinary differential equations with linear programming. Dynamical Systems, **17** (2002), 137–150.

[26] D. Norton, The fundamental theorem of dynamical systems, Comment. Math. Univ. Carolinae, **36-3** (1995), 585–597.

[27] Y. Oishi, Probabilistic design of a robust state-feedback controller based on parameter-dependent Lyapunov functions, In: Proceedings of the 42th IEEE Conference on Decision and Control (2003), 1920–1925.

[28] A. Papachristodoulou and S. Prajna, The construction of Lyapunov functions using the sum of squares decomposition. In: Proceedings of the 41st IEEE Conference on Decision and Control (2002), 3482–3487.

[29] P. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization". Ph.D. Thesis: Caltech, Pasadena, USA, 2000.

[30] M. Peet, Exponentially stable nonlinear systems have polynomial Lyapunov functions on bounded regions, IEEE Trans. Automatic Control **54** (2009), 979–987.

[31] M. Peet and A. Papachristodoulou, A converse sum-of-squares Lyapunov result: An existence proof based on the Picard iteration. In: 49th IEEE Conference on Decision and Control (2010), 5949–5954.

[32] S. Ratschan and Z. She Providing a Basin of Attraction to a Target Region of Polynomial Systems by Computation of Lyapunov-like Functions, SIAM J. Control and Optimization **48-7** (2010), 4377–4394.

[33] M. Rezaiee-Pajand and B. Moghaddasie, Estimating the Region of Attraction via collocation for autonomous nonlinear systems, Struct. Eng. Mech., **41-2** (2012), 263–284.

[34] C. Sanderson, "Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments", Technical Report, NICTA, 2010.

[35] S. Sastry, "Nonlinear Systems: Analysis, Stability, and Control", Springer, 1999.

[36] V. Zubov, "Methods of A. M. Lyapunov and their application", P. Noordhoff, Groningen, Netherlands, 1964.