

Stability verification for monotone systems using homotopy algorithms

Björn S. Rüffer Fabian R. Wirth

Received: date / Accepted: date

Abstract A monotone self-mapping of the nonnegative orthant induces a monotone discrete-time dynamical system which evolves on the same orthant. If with respect to this system the origin is attractive then there must exist points whose image under the monotone map is strictly smaller than the original point, in the component-wise partial ordering. Here it is shown how such points can be found numerically, leading to a recipe to compute order intervals that are contained in the region of attraction and where the monotone map acts essentially as a contraction. An important application is the numerical verification of so-called generalized small-gain conditions that appear in the stability theory of large-scale systems.

Mathematics Subject Classification (2000) 93C55 47H07 65H20

1 Introduction

By \mathbb{R}_+ we denote the nonnegative real numbers, $[0, \infty)$. A class \mathcal{K} function is a continuous function $\gamma: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ that satisfies $\gamma(0) = 0$ and is strictly increasing. The function γ is of class $\mathcal{K}_\infty \subset \mathcal{K}$ if in addition γ is unbounded. Note that with respect to composition the class \mathcal{K}_∞ is a group and the class \mathcal{K} a semi-group. Moreover, sums and positive multiples of \mathcal{K} functions are again \mathcal{K} functions.

The nonnegative orthant \mathbb{R}_+^n induces a partial order on \mathbb{R}^n , which coincides with the component wise ordering, and we write for $x, y \in \mathbb{R}^n$, $x \leq y$ if $y - x \in \mathbb{R}_+^n$, $x < y$ if $[x \leq y \text{ and } x \neq y]$, and $x \ll y$ if $y - x \in \text{int } \mathbb{R}_+^n$, the interior of \mathbb{R}_+^n .

B.S. Rüffer

Institut für Elektrotechnik und Informationstechnik, Universität Paderborn, Warburger Str. 100, 33098 Paderborn, Germany E-mail: bjorn@rueffer.info

F.R. Wirth

Institut für Mathematik, Universität Würzburg, Am Hubland, 97074 Würzburg, Germany E-mail: wirth@mathematik.uni-wuerzburg.de

A map $T: \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ is monotone if $x \leq y$ implies $Tx \leq Ty$. For any \mathcal{H}_∞ function ρ , the map $D = \text{diag}(\rho): \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ defined by $(Dx)_i = \rho(x_i)$ is an example of a monotone map.

We consider the following problem:

Problem 1 Let a monotone, continuous map $T: \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ with $T(0) = 0$ and a real number $r > 0$ be given. Find $s^* \in \mathbb{R}_+^n$ satisfying

1. $Ts^* \ll s^*$ (which implies $s^* \gg 0$),
2. $\|s^*\|_1 = r$, and
3. $Ts \not\leq s$ for all $s \in [0, s^*], s \neq 0$. □

The existence of such an s^* (for sufficiently small $r > 0$) is a necessary and sufficient condition for asymptotic stability of the origin with respect to the discrete time system

$$s^+ = Ts, \quad s \in \mathbb{R}_+^n. \quad (1)$$

It also arises as a so-called generalized small-gain condition [18]. If it is satisfied then the set $[0, s^*]$ is contained in the region of attraction. Moreover, in this case also the system with input,

$$s^+ = Ts + w, \quad s, w \in \mathbb{R}_+^n, \quad (2)$$

is *locally input-to-state stable* [8, 9], and knowledge of s^* yields estimates for the sets of admissible inputs and initial conditions which result in bounded outputs, cf. [18].

The numerical solution of this problem is interesting in two aspects: First of all, it provides a numerical way to make a qualitative assertion. Secondly, this assertion is not only of qualitative nature (i.e., a system is stable in some sense), but also quantitative in that an estimate for the region of attraction is obtained as well.

The numerical solution of this problem is interesting for several applications: In the context of large-scale interconnections of nonlinear systems the solution to Problem 1 can assert (local) input-to-state stability of interconnections of many systems in arbitrary interconnection topology. This in turn can be useful for formation control [19] or the effective implementation of decentralized model predictive control [15]. Furthermore, in the same context the knowledge of s^* can be used to find a locally Lipschitz continuous Lyapunov function for the composite large-scale system. Another application is in queuing theory. Here the solution to Problem 1 can be used to ascertain that a given switching policy stabilizes a flow switching network via the use of a monotone monodromy operator, cf. [7], at least for specified range of initial buffer levels and bounded inflow.

It is known that a solution to Problem 1 must exist for any $r > 0$ if the origin is globally attractive with respect to (1). In this case necessarily it holds that $Ts \not\leq s$ for all $s \in \mathbb{R}_+^n, s \neq 0$, and by virtue of a topological fixed point result, for every $r > 0$, there exists an s^* with $\|s^*\|_1 = r$ satisfying $Ts^* \ll s^*$. However, even if $T: \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ is monotone, continuous, and satisfies $T(0) = 0$ as well as $Ts \not\leq s$ for all $s \in \mathbb{R}_+^n, s \neq 0$, the origin is not necessarily *globally* attractive with respect to (1) (but it is so locally). For T of particular form a sufficient condition for global asymptotic stability of the origin with respect to (1) is the existence of a diagonal map $D = \text{diag}(\text{id} + \rho): \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ with $\rho \in \mathcal{H}_\infty$ such that for all $s \neq 0$, $(D \circ T)s \not\leq s$, or, equivalently $(T \circ D)s \not\leq s$. In this case the system (2) is input-to-state stable [16, 17].

Also known is that if T satisfies $T(s \oplus v) = Ts \oplus Tv$ for all $s, v \in \mathbb{R}_+^n$, where \oplus denotes component-wise maximization, then T must be of the form $(Ts)_i = \max_j \gamma_j(s_j)$ for all i, j , where $\gamma_j: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ are nondecreasing functions. In this case T has been termed *max-preserving*. Here, the condition $Ts \not\leq s$ for all $s \neq 0$ is equivalent to the *cycle condition*, which assumes that $\gamma_{1i_2} \circ \gamma_{2i_3} \circ \dots \circ \gamma_{k-1i_k} \circ \gamma_{ki_1} < \text{id}$ for all finite ordered sequences $(i_1, i_2, \dots, i_k) \subset \{1, \dots, n\}^k$. If this condition holds then with e denoting the vector $(1, \dots, 1)^T$ and $t > 0$ one has for $q(t) := \max\{te, T(te), \dots, T^{n-1}(te)\} \in \mathbb{R}_+^n$ a continuous path $q: \mathbb{R}_+ \rightarrow \mathbb{R}_+^n$ which is unbounded and nondecreasing in every component and satisfies $T(q(t)) \leq q(t)$. Re-parametrization yields a path $\tilde{q}(r)$ satisfying $\|\tilde{q}(r)\|_1 = r$, which can be interpreted as a parametrized “almost” solution to Problem 1, cf. [10].

In the linear case the action of T can be represented by multiplication with a nonnegative matrix (i.e., every component is nonnegative), which we also denote by $T \in \mathbb{R}_+^{n \times n}$. It is known that the following are equivalent:

1. $Ts \not\leq s$ for all $s \in \mathbb{R}_+^n, s > 0$;
2. there exists a $D = \text{diag}(\text{id} + \rho)$ with $\rho \in \mathcal{K}_\infty$ such that $(T \circ D)s \not\leq s$ for all $s \in \mathbb{R}_+^n, s > 0$ (equivalently, $(D \circ T)s \not\leq s$ for all $s \in \mathbb{R}_+^n, s > 0$);
3. the spectral radius of T is less than one;
4. there exists a unit vector $s^* \gg 0$ (with respect to the 1-norm) such that $Ts^* \ll s^*$, hence also the ray given by $rs^*, r \in \mathbb{R}_+$ satisfies $T(rs^*) \ll rs^*$ for $r > 0$ and $\|rs^*\|_1 = r$;
5. the inverse of $I - T$ exists and is given by the nonnegative matrix $\sum_{k \geq 0} T^k$.

The existence of the vector s^* is of course related to the classical Perron-Frobenius theory. If T is primitive then s^* is just the positive Perron-Frobenius root corresponding to the maximal eigenvalue, which coincides with the spectral radius. Further extensions of the classical Perron-Frobenius theory exist for special classes of nonlinear maps, in particular for homogeneous maps and for concave maps, cf. [1, 12].

In this paper we propose the use of a homotopy method to find a point s^* satisfying $Ts^* \ll s^*, \|s^*\|_1$ for any given $r > 0$. Our method of choice is the K1 algorithm proposed by Eaves [6] as a computational version of a topological fixed point theorem. There are, however, more elaborate choices of related algorithms available as well, cf. [2, 3] for an overview. In addition, there exist several skillful implementations of homotopy methods for solving polynomial systems; of these PHCpack [20], Bertini [5], and HOM4PS-2.0 [14] deserve mentioning. One particular advantage of homotopy methods is that they offer global convergence: If there exists a point s^* with $\|s^*\|_1 = r$ with the desired properties then it will be found. This has to be seen in contrast to methods based on Newton steps, which only guarantee convergence if the algorithm is started sufficiently close to s^* . Moreover, Newton methods usually assume some level of smoothness, whereas homotopy methods only require continuity.

Once the point s^* has been computed, the remaining verification of property 3 in Problem 1 is an easy task: It only needs to be checked that the sequence $T^k(s^*)$ converges to zero. It is then a consequence of monotonicity that property 3 must hold.

The paper is organized as follows. In Section 2 a few facts about monotone self-mappings of the nonnegative orthant and their induced discrete-time systems are re-

called. In particular, the topological fixed point theorem by Knaster, Kuratowski, and Mazurkiewicz is discussed. A brief and informal description of some of the underlying principles of Eaves' and other homotopy algorithms is given in Section 3. A MATLAB version of Eaves' K1 algorithm is provided in Section 4. Section 5 explains a short procedure to solve Problem 1 based on the use of a homotopy algorithm and the computation of one trajectory of system (1). This is followed by several numerical examples in Section 6.

2 Monotone maps and monotone discrete-time systems

This section collects a few theoretical results from the literature.

Throughout this section let $T: \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ be monotone and continuous with $T(0) = 0$. Consider also the induced discrete-time systems (1) and (2). Denote their respective solutions for initial condition $s^0 \in \mathbb{R}_+^n$ and, in case of (2), input sequence $w = \{w(k) \in \mathbb{R}_+^n : k \geq 0\}$, at time $k \geq 0$ by $\phi_{(1)}(k, s^0)$ and $\phi_{(2)}(k, s^0, w)$, respectively. If the reference to a particular system is clear from the context we omit the reference to the system. Observe that both systems satisfy the *ordering of solutions principle*: If $s^0 \leq v^0$ and for all $k \geq 0$, $w(k) \leq u(k)$ (which we abbreviate by $w \leq u$), then also $\phi_{(1)}(k, s^0) \leq \phi_{(1)}(k, v^0)$ and $\phi_{(2)}(k, s^0, w) \leq \phi_{(2)}(k, v^0, u)$ for all $k \geq 0$.

We denote the sphere in \mathbb{R}_+^n of radius $r > 0$ with respect to the 1-norm by $S_r = \{s \in \mathbb{R}_+^n : \|s\|_1 = \sum_i s_i = r\}$. Observe that S_r is an $(n-1)$ simplex.

Theorem 1 *Let $T: \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ be monotone and continuous with $T(0) = 0$. Assume that the origin is attractive with respect to (1) and denote the domain of attraction by \mathcal{B} . Then the following assertions hold:*

1. For every $s \in \mathcal{B}$, $s \neq 0$, necessarily $Ts \not\leq s$.
2. If $R \in \mathbb{R}$, $R > 0$ is such that $S_R \subset \mathcal{B}$ then for all $r \in (0, R]$ there exists a point $s \in S_r$, $s \gg 0$, satisfying $Ts \ll s$.
3. The origin is stable in the sense of Lyapunov, i.e., for every $\varepsilon > 0$ there exists a $\delta > 0$ such that $\|s\|_1 < \delta$ implies $\|Ts\|_1 < \varepsilon$.

A proof can be found in [16]. The first assertion is not difficult to prove directly, and the third follows from the second. The second assertion is the most technical one, and it is based on the covering theorem by Knaster, Kuratowski, and Mazurkiewicz (KKM) [11]. Our reasoning is based on the extension given in [13] which allows to consider coverings of open instead of closed sets. The argument is basically the following: One has $Ts \not\leq s$ for all $s \in S_r$, and S_r is a simplex. This ordering condition implies that the simplex is covered by the sets

$$\Omega_i = \{s \in S_r : (Ts)_i < s_i\}.$$

Moreover, no Ω_i can contain the face $\sigma_i \subset S_r$ opposite of the vertex $re_i \in \Omega_i$, where e_i denotes the i th unit vector. On the other hand, every k -dimensional simplex $\sigma = \text{co}\{re_i\}_{i \in \{i_1, \dots, i_k\}}$, $k \leq n-1$, is contained in the union $\bigcup_{i \in \{i_1, \dots, i_k\}} \Omega_i$. These are the prerequisites of the KKM Theorem, which then ascertains that the intersection $\bigcap \Omega_i$ must be nonempty. The proof of the KKM Theorem is based on the fact that every

simplicial refinement of S_r must contain at least one special simplex, which is again covered by all sets Ω_i but not by any strict subclass of $\{\Omega_1, \dots, \Omega_n\}$. As the size of the refinements tends to zero, this special simplex contracts to a point — the point of interest $s^* \in S_r$ satisfying $Ts^* \ll s^*$. In essence this proof technique relies on a fine discretization of S_r and an exhaustive search, which is not implementation friendly if n becomes large.

3 Homotopy based fixed point algorithms

An alternative proof of the KKM result has been given in [6] by means of a homotopy algorithm. In contrast to the original proof, here the initial simplex is successively refined in every iteration, thus the area that must contain s^* becomes smaller and smaller as iterations progress. We give a simplified account on the ideas behind this algorithm.

As in the original KKM paper, each point in $s \in S_r$ is assigned an integer label

$$l(s) = \max\{i: s \in \Omega_i\}. \quad (3)$$

Observe that $l(re_i) = i$ for $i = 1, \dots, n$. To keep things uncluttered, let us call a set of k distinct points (vertices) $\sigma = \{v^i \in \mathbb{R}_+^n\}_{i=1}^k$ a $(k-1)$ -set. The barycentric centre of such a $(k-1)$ -set is the point $v = \frac{1}{k} \sum_{i=1}^k v^i$. We call an $(n-1)$ -set σ *complete* if all vertices have distinct labels. Equivalently, every label $1, \dots, n$ gets assigned to a vertex exactly once. Observe that the $(n-1)$ -set $\sigma^0 = \{re_i\}_{i=1}^n$ is complete if $Ts \not\leq s$ for all $s \in S_r$.

Now one can also consider an n -set τ obtained from a given $(n-1)$ -set σ by adjoining one additional vertex v taken from the convex hull $\text{co}(\sigma)$ of the vertices of σ . For example, one could augment σ with its barycentric centre to obtain such an n -set τ .

The following observation is at the core of the fixed point algorithms by Eaves and also at the core of all related homotopy algorithms, cf. also [2, Thm. 1.7].

Theorem 2 *Every n -set has either none or exactly two complete $(n-1)$ -subsets.*

Proof Let $\tau = \{v^1, \dots, v^{n+1}\}$ denote the given n -set with its $n+1$ vertices. Now either $l(\{v^1, \dots, v^{n+1}\})$ contains the set $\{1, \dots, n\}$. In this case one label must get assigned twice, i.e., there exists a unique i and j, k such that $l(v^j) = l(v^k) = i$. In this case $\tau \setminus \{v^j\}$ and $\tau \setminus \{v^k\}$ are both complete $(n-1)$ -sets. All other $(n-1)$ -subsets must contain v^j and v^k and hence cannot be complete. And if $\{1, \dots, n\} \not\subseteq l(\{v^1, \dots, v^{n+1}\})$ then no such $(n-1)$ -set can exist. \square

The basic algorithm is now the following: $\sigma^0 = \{re_i\}_{i=1}^n$ serves as a complete *entry* $(n-1)$ -set. Successively, another vertex is added, say the barycentric centre v of σ^0 , to obtain the n -set $\tau = \sigma^0 \cup \{v\}$. By Theorem 2, τ contains exactly one $(n-1)$ -subset distinct from σ^0 , which we denote by σ^1 . Progressing inductively, one obtains a sequence of complete $(n-1)$ -sets σ^k whose area tends to zero.

The catch, however, is that the sequence does not necessarily contract to a point. Instead, σ^k may become “long and thin” as k becomes large. To prevent this kind

of behaviour, a more sophisticated choice of new vertices v is required. For this choice there exist a variety of alternatives. The paper by Eaves [6] proposes two such choices, which are named K1 and K2. Both guarantee convergence. Of these two choices K1 is the easiest and shortest to implement, and therefore we have chosen K1 for this exposition. On the other hand, it does not converge as quickly as K2, as has already been observed in [6]. It should be noted, however, that even more sophisticated pivoting strategies and restart algorithms can be found in [2–4] and the references contained therein. A detailed discussion of these is far beyond the scope of this paper. In principle any of these could have been used instead of our particular choice for Eaves’ K1 algorithm here.

4 Implementation of Eaves’ K1 algorithm

The integer labeling function (3) is numerically not feasible, instead we use the function

$$l_\varepsilon(s) = \max\{i: (Ts)_i + \varepsilon \leq s_i\}, \quad (4)$$

where $\varepsilon > 0$ is a design parameter. It guarantees that if a point s^* is obtained with the fixed point algorithm, then this point does in fact satisfy $Ts^* \ll s^*$ and not just $Ts^* < s^*$.

Obviously, if a point s^* with $Ts^* \ll s^*$ exists at all, then it has to be found by the algorithm if only $\varepsilon > 0$ is small enough. On the other hand, from practice it is fair to say that larger ε yield faster convergence (i.e., fewer iterations are necessary to obtain s^*).

Also it could be noted that instead of the maximal i in (4) the minimal or even any other unique choice should in theory do equally well. In particular, this may give rise to a different pivoting strategy by shifting the pivoting from the homotopy algorithm to the labeling function.

4.1 MATLAB code

Listing 1 Eaves’ algorithm based on the K1 complex [6] implemented in MATLAB.

```

1 function [kkmp, noit, succ] = eavesK1(monmap, r, n)
2     succ = 0; noit = Inf; global eavesK1_MAXREFINE
3     if isempty(eavesK1_MAXREFINE), eavesK1_MAXREFINE=1e3; end
4 % Initialization of the complex
5     q = -eye(n) + diag(ones(n-1,1),-1); v = zeros(n,n+1);
6     v(1,1)=1; v(n,1)=1; g=[n,1:n-1];
7     for i = 1 : n, v(:,i+1) = v(:,i) + q(:,g(i)); end
8 % compute initial labels
9     istar=1; inew=1; l=zeros(1,n+1);
10    for i = 1 : n+1,
11        x = v(:,i)/ norm(v(:,i),1) * r;
12        try
13            l(i) = eavesLabel(monmap,x);
14        catch exception,
15            error('initial_label_computation_went_wrong');

```

```

16     end
17 end
18 % the main loop
19     lexo=1:n+1;
20     for noit=1:eavesK1_MAXREFINE,
21 % identify vertex to drop and add new one
22         t=find(l==l(inew));
23         if t(1) == inew, istar=t(2); else istar=t(1); end
24         if istar==1, vnew=v(:,lexo(n+1))+q(:,g(1)); inew=n+1;
25             lexo=[lexo(2:n+1) lexo(1)]; v(:,lexo(inew))=vnew;
26             g=[g(2:n) g(1)]; l = [l(2:n+1) 0];
27         elseif istar < n+1,
28             vnew=v(:,lexo(istar-1))+q(:,g(istar));
29             inew=istar; lexo=lexo; v(:,lexo(inew))=vnew;
30             temp = g(istar); g(istar) = g(istar-1);
31             g(istar-1)=temp; clear temp;
32             l=[l(1:istar-1) 0 l(istar+1:n+1)];
33         else vnew=v(:,lexo(1))-q(:,g(n)); inew=1;
34             lexo=[lexo(n+1) lexo(1:n)]; v(:,lexo(inew))=vnew;
35             g=[g(n) g(1:n-1)]; l=[0 l(1:n)];
36         end
37 % compute label of new vertex:
38         kkmp = vnew/ norm(vnew,1) * r;
39         try
40             l(inew) = eavesLabel(monmap, kkmp);
41         catch exception,
42             error('successive_label_computation_went_wrong');
43         end
44         if all( feval(monmap, kkmp) < kkmp ),
45             succ = 1; % success!
46             return;
47         end
48     end % refinements
49 end % function eavesK1
50 function l = eavesLabel(monmap, x)
51     global eavesLabel_distance
52     if isempty(eavesLabel_distance), eavesLabel_distance = 1e-2;
53     end
54     Tx = feval(monmap, x);
55     t=find(Tx + eavesLabel_distance <= x); l=max(t);
56     if isempty(l), error(['Found_point_without_label._Try_' ...
57         'decreasing_eavesLabel_distance.']); end
58 end % function eavesLabel

```

Of the output arguments $kkmp$ denotes the vector $s^* \in \mathbb{R}_+^n$, $noit$ the number of iterations that have been consumed by the algorithm to find $kkmp$, and $succ$ is either 1 to denote that the algorithm was successful and $kkmp$ is a point of interest and 0 otherwise.

Of the input arguments $monmap$ is a function handle to the monotone map $T: \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ which satisfies $T(0) = 0$. The parameter r is the radius $r > 0$ of the sphere $S_r \subset \mathbb{R}_+^n$, where the algorithm tries to find s^* . Lastly, n denotes the dimension of \mathbb{R}_+^n .

There are two additional global variables that can be tweaked to modify the performance of the implementation. They are $eavesK1_MAXREFINE$ with a default value of

1000 and `eavesLabel_distance` with a default value of 10^{-2} . Their meaning is explained below.

4.2 Usage

To use the above MATLAB function, one has to implement the monotone map T of interest into a MATLAB function. An example of this is given in Listing 2.

Listing 2 A MATLAB implementation of the monotone map defined in Example 1, cf. Section 6.

```

1 function y = T(x)
2   y=zeros(size(x));
3   for k=1:length(x)-1,
4     y(k) = y(k) + x(k+1)^(k+1);
5     y(k+1) = y(k+1) + x(k)^(1/(k+1));
6   end
7   y=y/4;
8 end

```

Now, to compute a point $s^* \in \mathbb{R}_+^n$ satisfying $Ts^* \ll s^*$, $\|s^*\|_1 = r > 0$, one calls

```
[s_star, noit, succ] = eavesK1(@T, r, n)
```

which should yield the desired result. If necessary, the behaviour of the implementation can be fine-tuned by specifying

```
global eavesK1_MAXREFINE eavesLabel_distance
```

and assigning new values to these variables for the maximal number of iterations, and respectively, the parameter ε appearing in the labeling function (4).

4.3 Convergence

Assuming arbitrary precision computations and number representation as well as suitable choices for `eavesK1_MAXREFINE` (maximal number of iterations) and `eavesLabel_distance` (i.e., $\varepsilon > 0$ in (4)), the algorithm does what is expected:

Theorem 3 *Let $T: \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ be monotone and continuous. Let $r > 0$ and assume that $Ts \not\preceq s$ for all $s \in S_r$. Assume that there exists a point $s^* \in S_r$ such that $Ts^* \ll s^*$, $0 < \varepsilon < \min\{s_i^* - (Ts^*)_i\}$, where ε is the parameter in (4). Then the algorithm in Listing 1 produces a point $s^{**} \in S_r$ (possibly different from s^*) satisfying $Ts^{**} \ll s^{**}$, provided that the maximal number of allowed iterations is large enough.*

Proof The claim follows from the corresponding more general result in [6].

Note, however, that MATLAB itself does not offer arbitrary precision computations. In contrast, the package Bertini [5] uses arbitrary precision, while PHCPack [20] computes at a fixed high precision.

4.4 Remarks on computational complexity

In each iteration of the algorithm, the map $T: \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ has to be evaluated once to compute the label of the new vertex. In addition, n comparisons are necessary to find the old vertex with the same label as the new one to be dropped.

5 Algorithmic solution to Problem 1

Building upon an implementation as in the previous section, Problem 1 can now be solved as follows:

1. Given $r > 0$ compute s^* using the algorithm in the previous section, or using a more sophisticated implementation based on one of the algorithms proposed in e.g. [2–5, 14, 20]. If this is successful then properties 1 and 2 of Problem 1 are already satisfied.
2. Compute $\{T^k s^*\}_{k \geq 0}$. If this is a null-sequence then $Ts \not\leq s$ for all $s \in [0, s^*], s \neq 0$, i.e., property 3 of Problem 1 is satisfied.

The assertion of the second step in this short meta-algorithm relies on the first statement in Theorem 1. For if $T^k s^*$ tends to zero as $k \rightarrow \infty$ then $s^* \in \mathcal{B}$, the region of attraction. By the ordering of solutions principle also every point $s \in [0, s^*]$ must belong to \mathcal{B} as well.

The computational complexity of the second step consists of the computation of only one trajectory of a discrete-time monotone system. The trajectory has to be bounded, because due to the ordering of solutions principle it must be confined to the order interval $[0, s^*]$. Furthermore, due to the monotonicity of T it has to be non-increasing in every component, which allows to terminate further computation once $\phi_{(1)}(k, s^*)$ is sufficiently small for some $k \geq 0$.

Numerically it is impossible to check that the sequence defined by $s^k := T^k s^*$ is a null-sequence. In practice, one instead checks that for large N and small $\varepsilon > 0$ one has

$$\|s^N\| < \varepsilon.$$

Of course, this could lead to false positives, one should think. However, not all is lost, as we still recover a “practical” solution to the problem.

Proposition 1 *Let $T: \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ be monotone and continuous, let $\varepsilon > 0$, and let s^* be chosen such that $Ts^* \ll s^*$ and that for some $N \geq 1$, with $s_* := T^N s^*$, one has $\|s_*\| < \varepsilon$. Then the order interval $[0, s_*]$ is an asymptotically stable set with respect to system (1) and the set $\mathcal{B} := [0, s^*] \setminus [0, s_*]$ is contained in the basin of attraction. Necessarily it follows that $Ts \not\leq s$ for all $s \in \mathcal{B}$.*

In other words, while we cannot provide a “pure” solution to the third part of Problem 1 numerically, we can do so at least in a “practical” fashion.

Proof For any $s^0 \in \mathcal{B}$ it follows from monotonicity of T that $\phi_{(1)}(k, s^0) \leq \phi_{(1)}(k, s^*)$ and hence that $\phi_{(1)}(k, s^0) \leq s_*$ for all $k \geq N$. This shows that $[0, s_*]$ is an attractive set and \mathcal{B} in its basin of attraction. Stability follows with a similar argument. The algebraic condition $Ts \not\leq s$ for all $s \in \mathcal{B}$ is then easily derived using a contradiction argument.

6 Examples

In this section we consider a few numerical examples. The first is a nonlinear map $T: \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ which can be defined for any $n \geq 2$. For this map T it is known that

system (2) is input-to-state-stable, implying that the origin is globally asymptotically stable for system (1), and hence the Eaves K1 algorithm should produce an s^* for arbitrary $n \geq 2$ and $r > 0$.

The second example is a statistic generated from randomly chosen nonnegative matrices $A \in \mathbb{R}_+^{n \times n}$ with spectral radius less than one.

A third example shows that for a given $x(0)$ the pure iteration of $x(k+1) = Tx(k)$ does in general not produce an $x(k)$ such that $Tx(k) \ll x(k)$, even if $k \geq 0$ is large.

Example 1 Let $n \geq 2$. Consider the nonlinear map $T: \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ defined in [17, Example IV.1] given by

$$(Ts)_i = \frac{1}{4}(s_{i-1}^{1/i} + s_{i+1}^{i+1}), \quad s \in \mathbb{R}_+^n,$$

with the convention that $s_0 = s_{n+1} = 0$. For example, in the case $n = 5$ one has

$$Ts = \frac{1}{4} \begin{pmatrix} s_2^2 \\ \sqrt{s_1 + s_3^3} \\ \sqrt[3]{s_2 + s_4^4} \\ \sqrt[4]{s_3 + s_5^5} \\ \sqrt[5]{s_4} \end{pmatrix}.$$

Observe that $T(0) = 0$, and that T is obviously monotone and continuous. It has been shown in [17, Example IV.1] that the induced system (2) is input-to-state stable. Moreover, it can be verified directly that for every $r > 0$, the vector

$$p(r) = \begin{pmatrix} r \\ \sqrt{r} \\ \sqrt[3]{r} \\ \vdots \\ \sqrt[n]{r} \end{pmatrix}$$

satisfies $Tp(r) \ll p(r)$. Since $p: \mathbb{R}_+ \rightarrow \mathbb{R}_+^n$ is continuous, and in every component unbounded, for every $r > 0$ there must exist a point $s^* \in S_r$ such that $Ts^* \ll s^*$. Figure 1 shows how long it takes (in terms of iterations) for the Eaves' algorithm to find such a point s^* .

Example 2 It is relatively easy to generate many positive $n \times n$ matrices with a specified spectral radius in MATLAB. If the spectral radius $\rho(A)$ of a nonnegative matrix A is less than one then it defines a monotone mapping that should allow for a point $s^* \in S_r$ with $As^* \ll s^*$. Here we have chosen the spectral radius to be $\rho(A) = 0.8$ and have generated a number of matrices $A \in \mathbb{R}_+^{n \times n}$ for different choices of n . Again we have applied the algorithm in Listing 1 and counted the number of iterations needed. The outcome is plotted in Figure 2.

Example 3 Consider the monotone map $T: \mathbb{R}_+^2 \rightarrow \mathbb{R}_+^2$ given by $T(x) = (\sqrt{x_2}, \lambda x_1^2)^T$, with $\lambda \in (0, 1)$. Obviously $T(0) = 0$ and it is easy to check

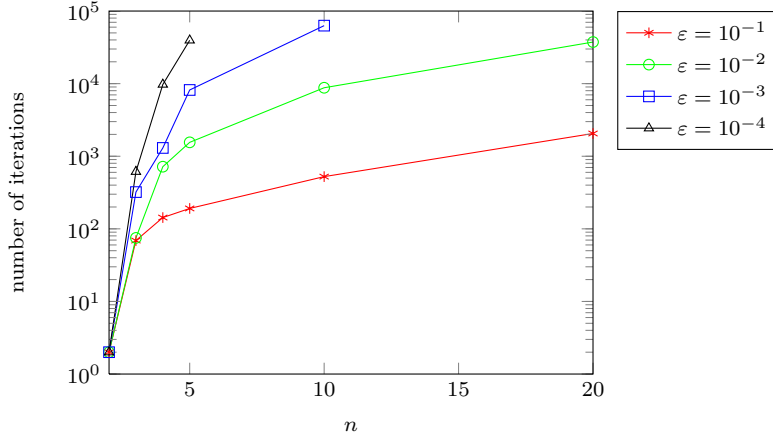


Fig. 1 Number of iterations needed by Eaves' K1 algorithm for $r = 10$, and different choices of n and $\varepsilon > 0$ in Example 1. The maximal number of iterations was set to 10^5 . It took 67 seconds to generate this whole plot on a 2.4 GHz Intel MacBook.

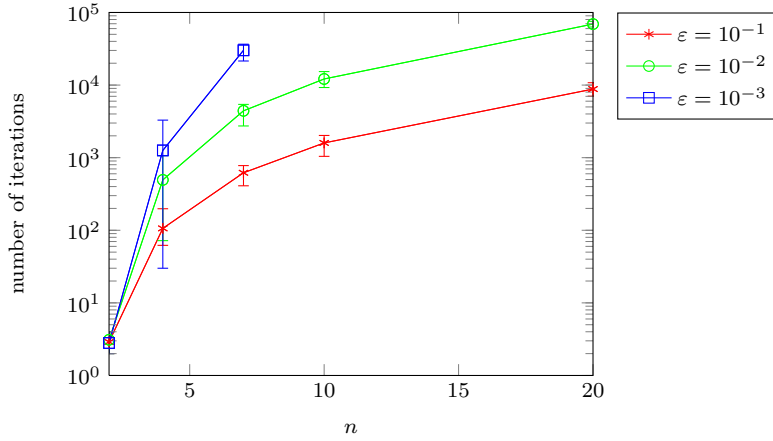


Fig. 2 Mean number of iterations needed for linear maps with spectral radius $\rho(A) = 0.8$ together with minimal and maximal number of iterations needed. Each data point corresponds to 10 randomly chosen matrices (using MATLAB's `rand(n)`). Generating this plot took 205 seconds (not taking into account generation of the test matrices). The maximal number of iterations was set to 10^5 .

that for any $x \in \mathbb{R}_+^n$, $T^k x \rightarrow 0$ as $k \rightarrow \infty$. Hence, T is a contraction. Yet, given $x \in \mathbb{R}_+^2$, if not already $Tx \ll x$ then there exists no $k \geq 1$ such that $T^{k+1}x \ll T^k x$. For we have,

$$T^2 x = \begin{pmatrix} \sqrt{\lambda} x_1 \\ \lambda x_2 \end{pmatrix}, \quad Tx = \begin{pmatrix} \sqrt{x_2} \\ \lambda x_1^2 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Assuming that $Tx \not\ll x$, we have either $(Tx)_1 = \sqrt{x_2} \geq x_1$, which implies $(T^2x)_2 = \lambda x_2 \geq \lambda x_1^2 = (Tx)_2$. Otherwise, we have $(Tx)_2 = \lambda x_1^2 \geq x_2$, implying that $(T^2x)_1 = \sqrt{\lambda} x_1 \geq \sqrt{x_2} = (Tx)_1$. This deduction repeats inductively. As a consequence, a pure iteration of the map T cannot yield a solution to Problem 1.

Acknowledgements B. S. Rüffer has been partially supported by the Australian Research Council's *Discovery Projects* funding scheme (project number DP0880494) and the Japan Society for the Promotion of Science. F. R. Wirth has been supported by the German Science Foundation (DFG) within the priority programme 1305: Control Theory of Digitally Networked Dynamical Systems. This work has been submitted while B. S. Rüffer was with the Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville VIC 3010, Australia.

The authors would like to thank Professor Sergey Dashkovskiy for numerous valuable discussions prior to this work. They would like to thank an anonymous reviewer for their constructive suggestions which helped to improve the presentation of the paper.

References

1. Dirk Aeyels and Patrick De Leenheer. Extension of the Perron-Frobenius theorem to homogeneous systems. *SIAM J. Control Optim.*, 41(2):563–582 (electronic), 2002.
2. Eugene Allgower and Kurt Georg. Simplicial and continuation methods for approximating fixed points and solutions to systems of equations. *SIAM Review*, 22:28–85, 1980.
3. Eugene L. Allgower and Kurt Georg. *Numerical continuation methods*, volume 13 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1990. An introduction.
4. Eugene L. Allgower and Kurt Georg. Numerical path following. In P. G. Ciarlet and J. L. Lions, editors, *Handbook of numerical analysis.*, volume V, pages 3–207. North-Holland, Amsterdam, 1997.
5. D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. Bertini: Software for numerical algebraic geometry. Available at <http://www.nd.edu/~sommese/bertini>.
6. B. C. Eaves. Homotopies for computation of fixed points. *Math. Programming*, 3:1–22, 1972.
7. V. Feoktistova and A. Matveev. Dynamic interactive stabilization of the switching Kumar-Seidman system. *Vestnik St. Petersburg Univ. Math.*, 42(3):226–234, 2009.
8. Kehan Gao and Yuandan Lin. On equivalent notions of input-to-state stability for nonlinear discrete time systems. In *Proc. of the IASTED Int. Conf. on Control and Applications*, pages 81–87, 2000.
9. Zhong-Ping Jiang and Yuan Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica J. IFAC*, 37(6):857–869, 2001.
10. Iasson Karafyllis and Zhong-Ping Jiang. A vector small-gain theorem for general nonlinear control systems. *Submitted to IEEE Trans. Automat. Control*, 2009. arXiv:0904.0755v1.
11. B. Knaster, C. Kuratowski, and S. Mazurkiewicz. Ein Beweis des Fixpunktsatzes für n -dimensionale Simplexe. *Fundamenta*, 14:132–137, 1929.
12. Ulrich Krause. Concave Perron-Frobenius theory and applications. *Nonlinear Anal.*, 47(3):1457–1466, 2001.
13. M. Lassonde. Sur le principe KKM. *C. R. Acad. Sci. Paris Sér. I Math.*, 310(7):573–576, 1990.

-
14. T. L. Lee, T. Y. Li, and C. H. Tsai. HOM4PS-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method. *Computing*, 83(2-3):109–133, 2008.
 15. D.M. Raimondo, L. Magni, and R. Scattolini. Decentralized MPC of nonlinear systems: An input-to-state stability approach. *International Journal of Robust and Nonlinear Control*, 17(17):1651–1667, 2007.
 16. B. S. Rüffer. Monotone inequalities, dynamical systems, and paths in the positive orthant of Euclidean n -space. *Positivity*, 14(2):257–283, June 2010.
 17. B. S. Rüffer. Small-gain conditions and the comparison principle. *IEEE Trans. Autom. Control*, 55(7):1732–1736, July 2010.
 18. B. S. Rüffer and S. N. Dashkovskiy. Local ISS of large-scale interconnections and estimates for stability regions. *Systems Control Lett.*, 59(3–4):241–247, 2010.
 19. H. G. Tanner, G. J. Pappas, and V. Kumar. Leader-to-formation stability. *IEEE Transactions on Robotics and Automation*, 20(3):443–455, 2004.
 20. J. Verschelde. Algorithm 795: PHCpack: a general-purpose solver for polynomial systems by homotopy continuation. *ACM Trans. Math. Softw.*, 25:251–276, June 1999.