Volume 26 Number 1

ACTA CYBERNETICA

Szeged, 2023

ACTA CYBERNETICA

Information for authors. Acta Cybernetica publishes only original papers in the field of Computer Science. Manuscripts must be written in good English. Contributions are accepted for review with the understanding that the same work has not been published elsewhere. Papers previously published in conference proceedings, digests, preprints are eligible for consideration provided that the author informs the Editor at the time of submission and that the papers have undergone substantial revision. If authors have used their own previously published material as a basis for a new submission, they are required to cite the previous work(s) and very clearly indicate how the new submission offers substantively novel or different contributions beyond those of the previously published work(s). There are no page charges. An electronic version of the published paper is provided for the authors in PDF format.

Manuscript Formatting Requirements. All submissions must include a title page with the following elements: title of the paper; author name(s) and affiliation; name, address and email of the corresponding author; an abstract clearly stating the nature and significance of the paper. Abstracts must not include mathematical expressions or bibliographic references.

References should appear in a separate bibliography at the end of the paper, with items in alphabetical order referred to by numerals in square brackets. Please prepare your submission as one single PostScript or PDF file including all elements of the manuscript (title page, main text, illustrations, bibliography, etc.).

When your paper is accepted for publication, you will be asked to upload the complete electronic version of your manuscript. For technical reasons we can only accept files in LaTeX format. It is advisable to prepare the manuscript following the guidelines described in the author kit available at `https://cyber.bibl.u-szeged.hu/index.php/actcybern/about/submissions` even at an early stage.

Submission and Review. Manuscripts must be submitted online using the editorial management system at `https://cyber.bibl.u-szeged.hu/index.php/actcybern/submission/wizard`. Each submission is peer-reviewed by at least two referees. The length of the review process depends on many factors such as the availability of an Editor and the time it takes to locate qualified reviewers. Usually, a review process takes 6 months to be completed.

Subscription Information. Acta Cybernetica is published by the Institute of Informatics, University of Szeged, Hungary. Each volume consists of four issues, two issues are published in a calendar year. Subscription rates for one issue are as follows: 5000 Ft within Hungary, €40 outside Hungary. Special rates for distributors and bulk orders are available upon request from the publisher. Printed issues are delivered by surface mail in Europe, and by air mail to overseas countries. Claims for missing issues are accepted within six months from the publication date. Please address all requests to:

Acta Cybernetica, Institute of Informatics, University of Szeged
P.O. Box 652, H-6701 Szeged, Hungary
Tel: +36 62 546 396, Fax: +36 62 546 397, Email: `acta@inf.u-szeged.hu`

Web access. The above information along with the contents of past and current issues are available at the Acta Cybernetica homepage `https://cyber.bibl.u-szeged.hu/` .

Tibor Gyimóthy
Department of Software Engineering, University of Szeged, Hungary
gyimothy@inf.u-szeged.hu

Zoltan Kato
Department of Image Processing and Computer Graphics, University of Szeged, Hungary
kato@inf.u-szeged.hu

Dragan Kukolj
RT-RK Institute of Computer Based Systems, Novi Sad, Serbia
dragan.kukolj@rt-rk.com

László Lovász
Department of Computer Science, Eötvös Loránd University, Budapest, Hungary
lovasz@cs.elte.hu

Kálmán Palágyi
Department of Image Processing and Computer Graphics, University of Szeged, Hungary
palagyi@inf.u-szeged.hu

Dana Petcu
Department of Computer Science, West University of Timisoara, Romania
petcu@info.uvt.ro

Andreas Rauh
School II – Department of Computing Science, Group Distributed Control in Interconnected Systems, Carl von Ossietzky Universität Oldenburg, Germany
andreas.rauh@uni-oldenburg.de

Heiko Vogler
Department of Computer Science, Dresden University of Technology, Germany
Heiko.Vogler@tu-dresden.de

Special Issue of the International Symposium on Scientific Computing, Computer Arithmetic and Verified Numerical Computation (SCAN)

Guest Editors

Andreas Rauh

Department of Computing Science,
University of Oldenburg, Germany
andreas.rauh@uni-oldenburg.de

Balázs Bánhelyi

Department of Computational Optimization,
University of Szeged, Hungary
banhelyi@inf.u-szeged.hu

Preface

The *19th International Symposium on Scientific Computing, Computer Arithmetic and Verified Numerical Computation (SCAN)* was originally planned to be organized by the Institute of Informatics of the University of Szeged (SZTE) in Szeged, Hungary, in the year 2020. Due to the pandemic situation, the Scientific Committee of SCAN decided to postpone the meeting to September 13–15, 2021 and to have it in a fully online format.

The members of the Scientific Committee were the following representatives of the topics of the conference: G. Alefeld (Karlsruhe, Germany), A. Bauer (Ljubljana, Slovenia), J. B. van den Berg (Amsterdam, the Netherlands), G.F. Corliss (Milwaukee, USA), T. Csendes (Szeged, Hungary), R.B. Kearfott (Lafayette, USA), V. Kreinovich (El Paso, USA), J.-P. Lessard (Montreal, Canada), W. Luther (Duisburg, Germany), S. Markov (Sofia, Bulgaria), G. Mayer (Rostock, Germany), J.-M. Muller (Lyon, France), M. Nakao (Tokyo, Japan), T. Ogita (Tokyo, Japan), S. Oishi (Tokyo, Japan), K. Ozaki (Tokyo, Japan), M. Plum (Karlsruhe, Germany), A. Rauh (Brest, France), N. Revol (Lyon, France), J. Rohn (Prague, Czech Republic), S. Rump (Hamburg, Germany/Tokyo, Japan), S. Shary (Novosibirsk, Russia), W. Tucker (Uppsala, Sweden), W. Walter (Dresden, Germany), J. Wolff von Gudenberg (Würzburg, Germany), and N. Yamamoto (Tokyo, Japan). The members of the Organizing Committee were: Balázs Bánhelyi, Tibor Csendes, Boglárka G.-Tóth, Viktor Homolya, Tamás Vinkó, and Dániel Zombori.

During SCAN, more than 50 participants were present and 48 talks in several fields of reliable computation and its applications were given, and organized in 18 thematic sessions. The plenary speakers were Fabienne Jézéquel (Sorbonne University, France), Marko Lange (Hamburg University of Technology, Germany), J.D. Mireles James (Florida Atlantic University, USA), together with the Moore Prize winners Marko Lange and Siegfried Rump (Waseda University, Japan).

The open-access scientific journal Acta Cybernetica offered to publish paper versions of selected presentations after a careful peer review process. Altogether, 7 papers were accepted for publication in the present special issue of Acta Cybernetica. The full program of the conference, the collection of all abstracts, and further information can be found at https://www.inf.u-szeged.hu/scan2020/.

Andreas Rauh and Balázs Bánhelyi
Guest Editors

Acta Cybernetica **26** (2023) 5–34.

Proving the Stability of the Rolling Navigation

Auguste Bourgois[a], Amine Chaabouni[b],
Andreas Rauh[c], and Luc Jaulin[d]

Abstract

In this paper, we propose to study the stability of a navigation method that allows a robot to move in an unstructured environment without compass by measuring a scalar function φ which only depends on the position. The principle is to ask the robot to *roll* along an isovalue of φ. Using an interval method, we prove the stability of our closed loop system in the special case where φ is linear.

Keywords: interval analysis, hybrid systems, stability

1 Introduction

The rolling navigation has first been presented in [34] in the context of a small flying drone following the border of a cloud. The only exteroceptive information the robot has is if it is inside or outside the cloud. Experimentally, the control strategy has been proved to be very robust even if we do not know the prior shape of the cloud. The closed loop system corresponds to a nonlinear hybrid system and the theoretical analysis of the stability is considered as difficult.

The goal of this paper is to show that interval-based methods [22] can be used to provide a rigorous stability analysis of such a hybrid dynamical system. Interval analysis has indeed been used to solve numerous practical problems (see *e.g.*, [16] for solving nonlinear problems, [30, 31] for localization and mapping, [9] for autonomous driving). In the context of dynamical systems and stability analysis, Tucker [33] has used interval analysis to prove that the Lorenz attractor exists and efficient solvers (such as CAPD) have been proposed for integrating differential equations [14, 35] in a rigorous way. The corresponding methods can then be used for stability analysis of nonlinear systems [5, 17, 28]. In the context of hybrid systems, even if guaranteed integration has been used for characterizing reachability

[a]FORSSEA Robotics, Paris, France, E-mail: auguste@forssea-robotics.fr, ORCID: 0000-0002-0333-5872

[b]Ecole Polytechnique, Palaiseau, France, E-mail: amine.chaabouni@polytechnique.edu

[c]Department of Computing Science, University of Oldenburg, Germany, E-mail: andreas.rauh@uni-oldenburg.de, ORCID: 0000-0002-1548-6547

[d]ENSTA-Bretagne, Brest, France, E-mail: lucjaulin@gmail.com, ORCID: 0000-0002-0938-0615

DOI: 10.14232/actacyb.295941

sets [24, 25], to our knowledge, it has never been used to check the stability of dynamical systems where jumps could occur.

The work is an extended version of the abstract presented for SCAN 2020 [6] which deals with the rigorous stability of hybrid systems. The main contribution of our paper is to propose a method which combines Poincaré maps with interval analysis in order to provide an attraction basin [2, 12, 19, 26] associated with an hydrid system. More precisely, we want to find a subset of the state space which will converge to a stable periodic orbit.

Section 2 recalls the basic definitions related to Poincaré maps and stability analysis of nonlinear discrete-time systems. Section 3 formalizes the problem of the rolling-based navigation of robots in terms of hybrid systems. Section 4 proves the stability of the rolling-based navigation in the case where the environment is linear. Section 5 shows how the stability analysis can be used to compute a set of initial vectors which will converge to a stable attractor and Section 6 concludes the paper.

2 Mathematical tools

In this section, we give the basic tools that will be used to prove the stability of a periodic orbit of a hybrid system.

2.1 Discrete-time positive invariant set

Consider the discrete-time system

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) \tag{1}$$

with $\mathbf{f}(\mathbf{0}) = \mathbf{0}$. A set $\mathbb{A}$ is positive invariant if $\mathbf{f}(\mathbb{A}) \subset \mathbb{A}$. We consider two types of sets for positive invariance: Ellipsoids or boxes.

Ellipsoid. Figure 1 (a) illustrates the case of a discrete-time system where the arrows represent the function $\mathbf{f}$. Subfigure (b) gives a positive invariant set.

To find such an ellipsoid $\mathcal{E}_{\mathbf{x}} : \mathbf{x}^{\mathrm{T}} \cdot \mathbf{P} \cdot \mathbf{x} \leq \varepsilon$, we can use the Lyapunov method in the linear case. If the system is stable and linear, we have

$$\mathbf{x}_{k+1} = \mathbf{A} \cdot \mathbf{x}_k \tag{2}$$

and we can find a positive definite matrix $\mathbf{P}$ (denoted by $\mathbf{P} \succ 0$) such that $V(\mathbf{x}) = \mathbf{x}^{\mathrm{T}} \cdot \mathbf{P} \cdot \mathbf{x}$ is a Lyapunov function:

$$\begin{aligned} & V(\mathbf{x}_{k+1}) = V(\mathbf{x}_k) - \mathbf{x}_k^{\mathrm{T}} \mathbf{x}_k \\ \Leftrightarrow \quad & \mathbf{x}_{k+1}^{\mathrm{T}} \cdot \mathbf{P} \cdot \mathbf{x}_{k+1} = \mathbf{x}_k^{\mathrm{T}} \cdot \mathbf{P} \cdot \mathbf{x}_k - \mathbf{x}_k^{\mathrm{T}} \mathbf{x}_k \\ \Leftrightarrow \quad & \mathbf{x}_k^{\mathrm{T}} \cdot \mathbf{A}^{\mathrm{T}} \cdot \mathbf{P} \cdot \mathbf{A} \cdot \mathbf{x}_k - \mathbf{x}_k^{\mathrm{T}} \cdot \mathbf{P} \cdot \mathbf{x}_k = -\mathbf{x}_k^{\mathrm{T}} \mathbf{x}_k \\ \Leftrightarrow \quad & \mathbf{x}_k^{\mathrm{T}} \cdot \left(\mathbf{A}^{\mathrm{T}} \cdot \mathbf{P} \cdot \mathbf{A} - \mathbf{P} \right) \cdot \mathbf{x}_k = -\mathbf{x}_k^{\mathrm{T}} \mathbf{x}_k. \end{aligned} \tag{3}$$

To determine the ellipsoid $\mathcal{E}_{\mathbf{x}}$, we have to solve the Lyapunov equation:

$$\mathbf{A}^{\mathrm{T}} \cdot \mathbf{P} \cdot \mathbf{A} - \mathbf{P} = -\mathbf{I} \tag{4}$$

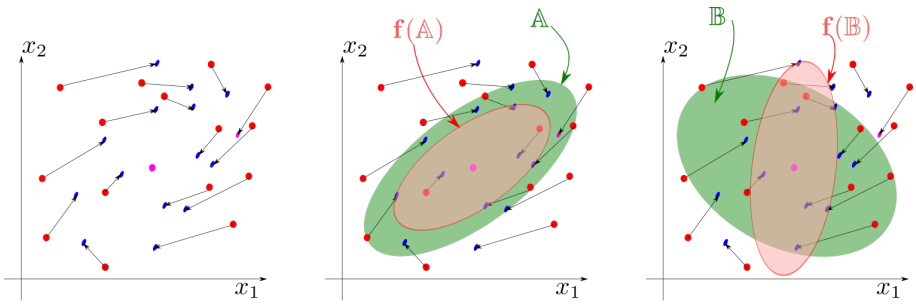


Figure 1: (a) A discrete-time system; (b) The green ellipse $\mathbb{A}$ is positive invariant; (c) The green ellipse $\mathbb{B}$ is not positive invariant

which is linear in $\mathbf{P}$. If the system is nonlinear, we apply the Lyapunov method on the linearized system and check the positive invariance using interval analysis [28]. There, an ellipsoidal calculus method is presented that allows for finding domains that certainly belong to the region of attraction of a stable equilibrium. Due to the straightforward implementation of these approaches, they can serve as an initialization of advanced procedures for finding the maximal provable attraction domains of nonlinear systems for which advanced methods based on linear matrix inequality or Bezoutian approaches were developed in [13, 23]. For methods, optimizing the quadratic Lyapunov functions for a stability proof of nonlinear autonomous systems using interval analysis, see [32]. Combinations of these methods with the box-type representation of invariant sets used in the remainder of this paper can be investigated in future work.

Boxes. To find a box which is positive invariant, we may use the centered form method [8, 28]. For this, check if

$$[\mathbf{J_f}]([\mathbf{x}]) \cdot [\mathbf{x}] \subset [\mathbf{x}] \tag{5}$$

where $\mathbf{J_f}(\mathbf{x})$ is the Jacobian matrix of $\mathbf{f}$ at $\mathbf{x}$ and $[\mathbf{J_f}]$ is its interval extension [22]. In some situations, such a box $[\mathbf{x}]$ does not exist. Now, for $k \in \{1, 2, \dots\}$, we have

$$\begin{array}{rcl} \mathbf{J}_{\mathbf{f}^k} & = & (\mathbf{J_f}(\mathbf{f}^{k-1})) \cdot \mathbf{J}_{\mathbf{f}^{k-1}} \\ \mathbf{f}^k & = & \mathbf{f} \circ \mathbf{f}^{k-1}. \end{array} \tag{6}$$

In this case, we search for the smallest k such that

$$[\mathbf{J}_{\mathbf{f}^k}]([\mathbf{x}]) \cdot [\mathbf{x}] \subset [\mathbf{x}], \tag{7}$$

where

$$\begin{array}{rcl} [\mathbf{J}_{\mathbf{f}^k}]([\mathbf{x}]) & = & [\mathbf{J_f}]([\mathbf{f}^{k-1}]([\mathbf{x}])) \cdot [\mathbf{J}_{\mathbf{f}^{k-1}}]([\mathbf{x}]) \\ [\mathbf{f}^k]([\mathbf{x}]) & = & [\mathbf{f}] \circ [\mathbf{f}^{k-1}]([\mathbf{x}]) \end{array} \tag{8}$$

as illustrated by Figure 2.

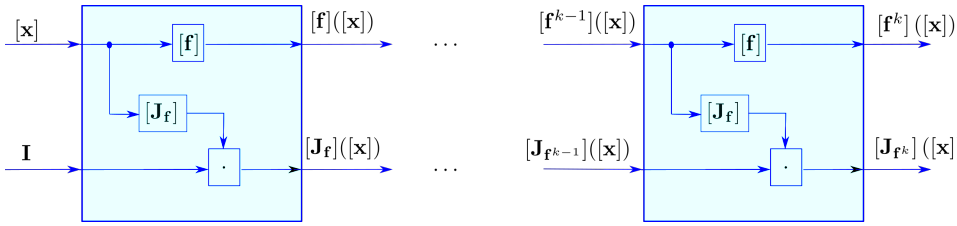


Figure 2: Sequential computation of $[\mathbf{f}^k]([\mathbf{x}])$ and $[\mathbf{J}_{\mathbf{f}^k}]([\mathbf{x}])$

For instance, for $k = 2$, we have to check that

$$\underbrace{([\mathbf{J_f}]([\mathbf{f}]([\mathbf{x}]))) \cdot [\mathbf{J_f}]([\mathbf{x}])}_{[\mathbf{J}_{\mathbf{f}^2}]([\mathbf{x}])} \cdot [\mathbf{x}] \subset [\mathbf{x}]. \tag{9}$$

For both approaches, we need to have an interval extension for $\mathbf{f}$ and for its Jacobian matrix $\mathbf{J_f}$. It is not a problem when we have an analytical expression for $\mathbf{f}$ but this is not always the case as we will see in the following section.

2.2 Poincaré map

Consider now a continuous-time system of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$, such as the Van der Pol system illustrated by Figure 3 which contains a stable periodic orbit γ. To prove the stability of γ, we use the Poincaré method. For this, we choose a point $\mathbf{x}_0 \in \gamma$. Then, we chose a $n-1$ dimensional manifold $\mathcal{S}$ called the *Poincaré section*. The Poincaré section $\mathcal{S}$ is chosen transversal to the flow of the system. It is such that $\mathcal{S} \cap \gamma = \{\mathbf{x}_0\}$. We assume that the points of $\mathcal{S}$ all satisfy the equation $g(\mathbf{x}) = 0$.

Assume that we have a *Cartesian parametrization* for $\mathcal{S}$, *i.e.*, a diffeomorphism $\mathbf{h} : \mathcal{A} \mapsto \mathcal{S}$ such that $\mathbf{h}(\mathbf{0}) = \mathbf{x}_0$, where $\mathcal{A} = \mathbf{h}^{-1}(\mathcal{S}) \subset \mathbb{R}^{n-1}$. The function $\mathbf{h}$ is called the *chart* for $\mathcal{S}$. It allows us to fix a coordinate system on $\mathcal{S}$. Equivalently, when $\mathbf{a}$ scans $\mathcal{A}$, $\mathbf{h}(\mathbf{a})$ scans $\mathcal{S}$.

We define the Poincaré map by:

$$\mathbf{p} : \begin{array}{ccc} \mathcal{A} & \to & \mathcal{A} \\ \mathbf{a} & \mapsto & \mathbf{p}(\mathbf{a}) \end{array} \tag{10}$$

where $\mathbf{p}(\mathbf{a})$ is the point in $\mathcal{A} \subset \mathbb{R}^{n-1}$ such that the trajectory initialized at $\mathbf{x}_a = \mathbf{h}(\mathbf{a}) \in \mathbb{R}^n$ intersects $\mathcal{S}$ for the first time at $\mathbf{x}_b = \mathbf{h}(\mathbf{p}(\mathbf{a}))$. Then we define the discrete-time system

$$\mathbf{a}(k+1) = \mathbf{p}(\mathbf{a}(k)). \tag{11}$$

If the sequence is asymptotically stable, then γ is an attractor of the vector field $\mathbf{f}$. Equivalently, we will say that γ is stable.

Now, the asymptotic stability of the Poincaré map $\mathbf{p}$ can be proved using the Lyapunov method, as described in the previous section, combined with interval

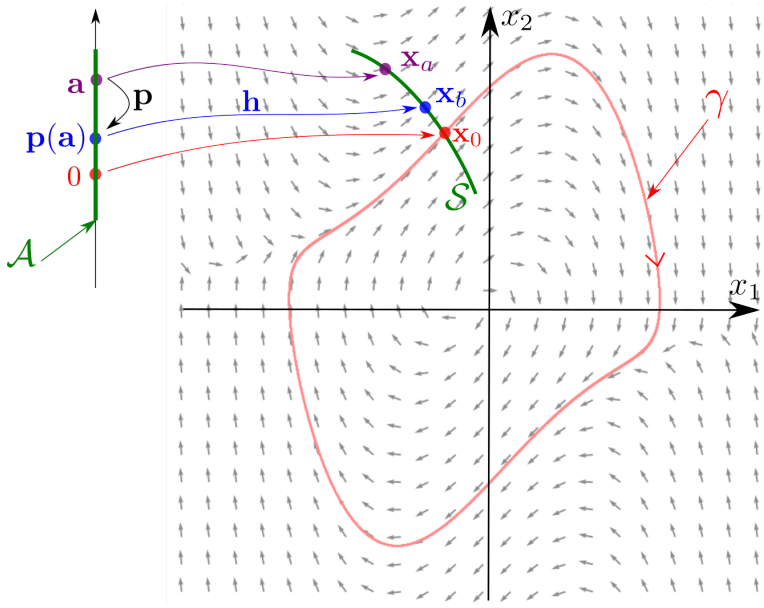


Figure 3: Stable periodic orbit (red) of a continuous time system

tools [29]. A three dimensional illustration of the Poincaré map is given in Figure 4. In the picture, both $\mathbf{a}$ and $\mathbf{p}(\mathbf{a})$ are represented in the 3D frame, at the places of $\mathbf{x}_a$ and $\mathbf{x}_b$, but they actually belong to the Cartesian plane $\mathbb{R}^2$ represented by the red base. For simplicity, we may confuse the part of the hyperplane $\mathcal{A} \subset \mathbb{R}^{n-1}$ and the surface $\mathcal{S}$.

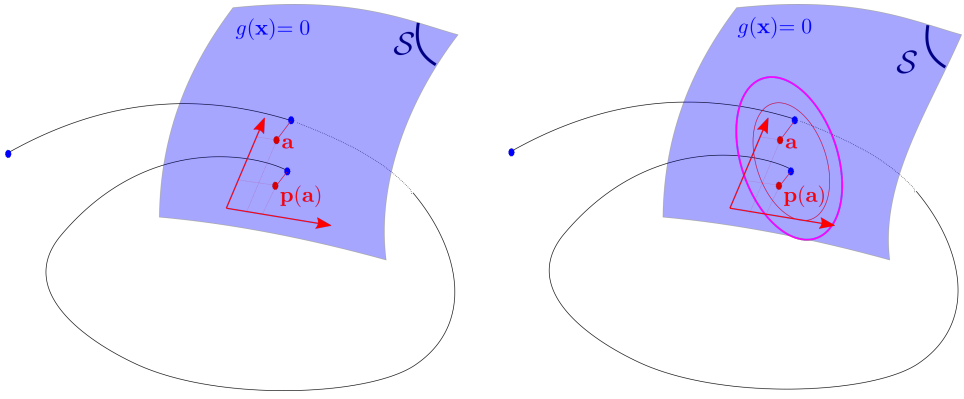


Figure 4: Poincaré map (left); Positive invariant ellipsoid of the Poincaré section (right)

2.3 Partial Poincaré maps

Hybrid systems [21], is a class of dynamical systems with discrete state q (for instance $q \in \{0, 1\}$) and a continuous state $\mathbf{x} \in \mathbb{R}^n$. The continuous state $\mathbf{x}$ follows a state equation of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, q).$$

When some equality conditions are satisfied for $\mathbf{x}$, say $g(\mathbf{x}, q) = 0$ then q may jump from one discrete state to another (*e.g.*, from q =0 to $q = 1$). The state vector $\mathbf{x}$ may jump also. For hybrid systems, we need more than one section to prove the stability [11]. An illustrative example will be given in Section 4. Since we have several sections, we will have several Poincaré maps. They will be called *partial* Poincaré maps. These maps also exist for dynamical systems that are not hybrid, but they are not strictly needed.

Figure 5 represents a situation with two sections $\mathcal{S}_1$, $\mathcal{S}_2$ and the partial Poincaré map is $\mathbf{p} : \mathcal{S}_1 \mapsto \mathcal{S}_2$ which is defined from a state equation of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$. We assume that it is possible to get a Cartesian frame for $\mathcal{S}_1$ and $\mathcal{S}_2$ thanks to charts $\mathbf{h}_1$ and $\mathbf{h}_2$. Using these charts we can define boxes on these sections.

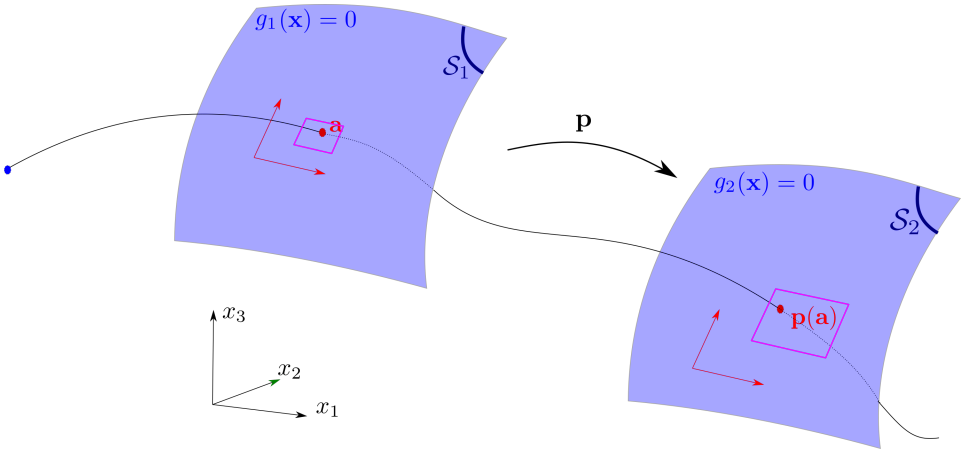


Figure 5: Partial Poincaré map to go from one section to another

2.4 Interval extension for the Poincaré map

Consider the system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \tag{12}$$

where $\mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^n$ is C^1 and Lipschitz continuous. The flow of the system is denoted by $\mathbf{\Phi}(\mathbf{x}_0, t)$. Take two Poincaré sections $\mathcal{S}_1, \mathcal{S}_2$ with charts $\mathbf{h}_1, \mathbf{h}_2$. We assume that $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$ and any trajectory initialized in $\mathcal{S}_1$, will cross $\mathcal{S}_2$ later. Define the two sets $\mathcal{A}_1 = \mathbf{h}_1^{-1}(\mathcal{S}_1) \subset \mathbb{R}^{n-1}$ and $\mathcal{A}_2 = \mathbf{h}_2^{-1}(\mathcal{S}_2) \subset \mathbb{R}^{n-1}$. Take a point $\mathbf{a} \in \mathcal{A}_1$ and denote by $\mathbf{x}_a = \mathbf{h}_1(\mathbf{a})$ the corresponding state vector in $\mathcal{S}_1 \subset \mathbb{R}^n$.

The trajectory initialized at $\mathbf{x}_a$ will cross $\mathcal{S}_2$ for the first time at $\mathbf{x}_b \in \mathbb{R}^n$ at time τ_b. Define by $\mathbf{b} = \mathbf{h}_2^{-1}(\mathbf{x}_b) \in \mathcal{A}_2$. Note that $\mathbf{a}$ and $\mathbf{x}_a$ correspond to the same quantity except that $\mathbf{a} \in \mathcal{A}_1$ whereas $\mathbf{x}_a \in \mathcal{S}_1 \subset \mathbb{R}^n$. The same remark could be done for the pair $(\mathbf{b}, \mathbf{x}_b)$. The Poincaré map $\mathbf{b} = \mathbf{p}(\mathbf{a})$ is here defined as the following composition:

$$\mathbf{a} \in \mathcal{A}_1 \subset \mathbb{R}^{n-1} \mapsto \mathbf{x}_a \in \mathcal{S}_1 \subset \mathbb{R}^n \mapsto \mathbf{x}_b = \mathbf{\Phi}(\mathbf{x}_a, \tau_b) \in \mathcal{S}_2 \subset \mathbb{R}^n \mapsto \mathbf{b} \in \mathcal{A}_2 \subset \mathbb{R}^{n-1}.$$

We want an interval extension for $\mathbf{p}$. We propose the following algorithm with the illustrating Figure 6.

Step 1. Take a box $[\mathbf{a}] \subset \mathcal{A}_1 \subset \mathbb{R}^{n-1}$ and compute $[\mathbf{x}_a] = [\mathbf{h}_1]([\mathbf{a}])$, where $[\mathbf{h}_1]$ is the interval extension of the chart $\mathbf{h}_1$.

Step 2. Integrate $[\mathbf{x}_a]$ to get a tube $[\mathbf{x}](\cdot)$ of $\mathbb{R}^n$. Note that $[\mathbf{x}](0) = [\mathbf{x}_a]$

Step 3. We compute the tubes $[y](\cdot) = [g_2]([\mathbf{x}](\cdot))$ and the tube $[\dot{y}](\cdot) = [\frac{\partial g_2}{\partial \mathbf{x}}] \cdot [\mathbf{f}]([\mathbf{x}](\cdot))$ and we select an interval $[t_1, t_2]$ which satisfies

$$\begin{array}{ll} (i) & [y]([0, t_1]) \subset]0, \infty] \\ (ii) & [y]([t_2]) \subset [-\infty, 0[\\ (iii) & [\dot{y}]([t_1, t_2])) \subset [-\infty, 0[\end{array} \tag{13}$$

If we fail to find this interval, we return a failure.

Step 4. Select the subtube $[\mathbf{x}]([t_1, t_2])$.

Step 5. Return a box $[\mathbf{x}_b] \subset \mathbb{R}^n$ which encloses the subtube $\cup_{t \in [t_1, t_2]} [\mathbf{x}](t)$.

Step 6. Compute a box $[\mathbf{b}]$ which encloses $\mathbf{h}_2^{-1}([\mathbf{x}_b] \cap \mathcal{S}_2)$

2.5 Variational equation

Consider again the system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}),\ \mathbf{x} \in \mathbb{R}^n \tag{14}$$

where $\mathbf{f}$ is now assumed to be twice differentiable. Denote by $\mathbf{\Phi}(\mathbf{x}_0, t)$ the flow for an initial vector $\mathbf{x}_0$. We define the *variational matrix* $\mathbf{J}(\mathbf{x}_0, t) = \frac{\partial \mathbf{\Phi}(\mathbf{x}_0, t)}{\partial \mathbf{x}_0}$. It describes the effect of a small perturbation on a given trajectory, while we make a small variation on the initial state vector $\mathbf{x}_0$. It can be shown that it satisfies the *variational equation* [1]

$$\dot{\mathbf{J}} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \cdot \mathbf{J}, \tag{15}$$

with $\mathbf{J}(0) = \mathbf{I}$ for which further applications are discussed with respect to sensitivity analysis and control design in [27]. Using an interval ODE solver, we get an enclosure for $\mathbf{x}(t)$ and $\mathbf{J}(t)$, for a given initial box $[\mathbf{x}_0]$.

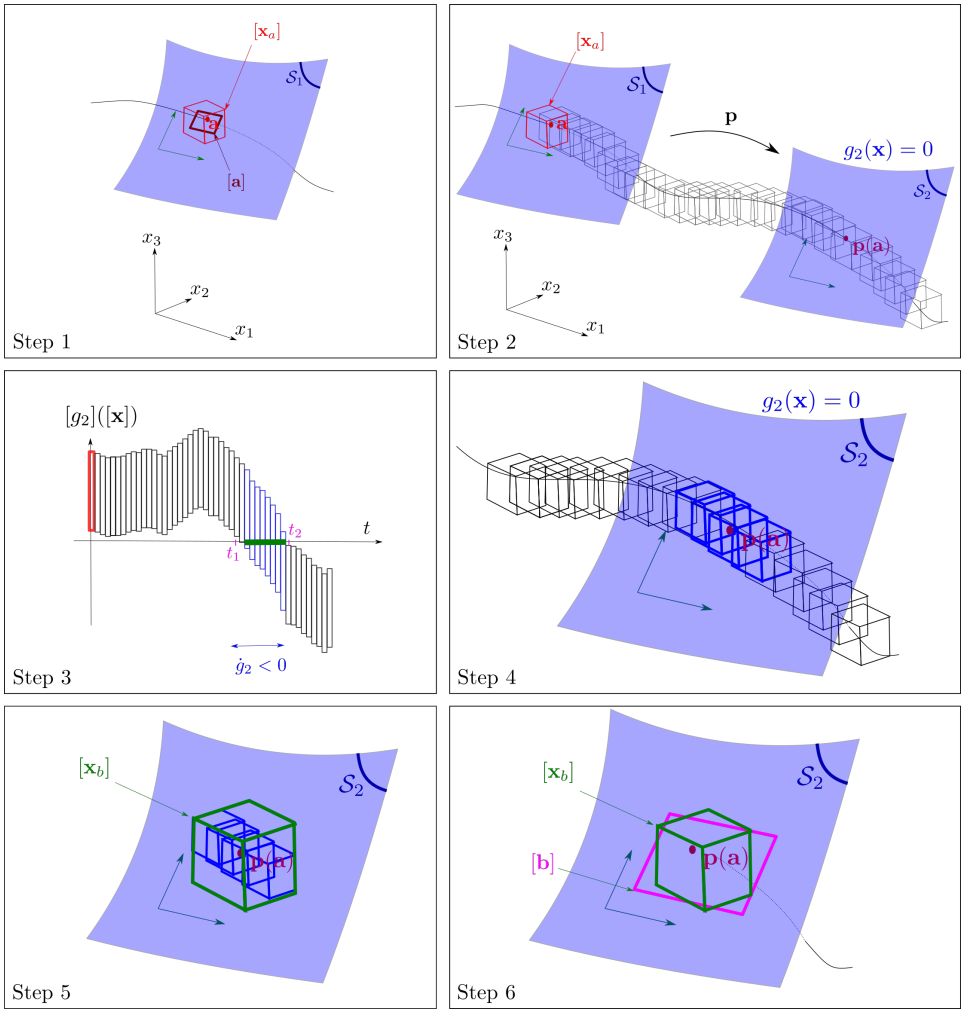


Figure 6: Poincaré interval extension

2.6 Jacobian of the Poincaré map

Proposition 1. *Consider a state equation with a flow* $\mathbf{\Phi}(\mathbf{x}, t)$ *and two Poincaré sections* $\mathcal{S}_1$, $\mathcal{S}_2$ *with equations* $g_1(\mathbf{x}) = 0$, $g_2(\mathbf{x}) = 0$. *The associated charts for* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *are denoted by* $\mathbf{h}_1, \mathbf{h}_2$. *Define* $\mathcal{A}_1 = \mathbf{h}_1^{-1}(\mathcal{S}_1)$ *and* $\mathcal{A}_2 = \mathbf{h}_2^{-1}(\mathcal{S}_2)$. *Denote by* $\mathbf{p} : \mathcal{A}_1 \to \mathcal{A}_2$ *the associated partial Poincaré map and by*

$$\tau(\mathbf{x}) = \min\{t > 0 | \mathbf{\Phi}(\mathbf{x}, t) \in \mathcal{S}_2\} \tag{16}$$

the associated Poincaré time function, which is assumed to exist for all $\mathbf{x} \in \mathcal{A}_1$. *Take* $\mathbf{a} \in \mathcal{A}_1$. *We have*

$$\mathbf{J}_{\mathbf{p}}(\mathbf{a}) = \frac{\partial \mathbf{p}}{\partial \mathbf{a}}(\mathbf{a}) = \frac{\partial \mathbf{h}_2^{-1}}{\partial \mathbf{x}}(\mathbf{x}_b) \cdot \frac{\partial \mathbf{q}}{\partial \mathbf{x}}(\mathbf{x}_a) \cdot \frac{\partial \mathbf{h}_1}{\partial \mathbf{a}}(\mathbf{a}), \tag{17}$$

where

$$\begin{array}{lrcl} (i) & \mathbf{q}(\mathbf{x}_a) & = & \mathbf{\Phi}(\mathbf{x}_a, \tau(\mathbf{x}_a)) \\ (ii) & \frac{\partial \mathbf{q}}{\partial \mathbf{x}}(\mathbf{x}_a) & = & \mathbf{J}_m + \mathbf{f}(\mathbf{x}_b) \cdot \frac{\partial \tau}{\partial \mathbf{x}}(\mathbf{x}_a) \\ (iii) & \frac{\partial \tau}{\partial \mathbf{x}}(\mathbf{x}_a) & = & -\frac{1}{\frac{\partial g_2}{\partial \mathbf{x}}(\mathbf{x}_b) \cdot \mathbf{f}(\mathbf{x}_b)} \cdot \frac{\partial g_2}{\partial \mathbf{x}}(\mathbf{x}_b) \cdot \mathbf{J}_m \end{array} \tag{18}$$

and

$$\begin{array}{rcl} \mathbf{x}_a & = & \mathbf{h}_1(\mathbf{a}), \mathbf{a} \in \mathbb{R}^{n-1} \\ \mathbf{x}_b & = & \mathbf{h}_2(\mathbf{b}), \mathbf{b} \in \mathbb{R}^{n-1} \\ \mathbf{b} & = & \mathbf{p}(\mathbf{a}) \\ \mathbf{J}_m & = & \frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}}(\mathbf{x}_a, \tau(\mathbf{x}_a)) \qquad \textit{(monodromy matrix).} \end{array} \tag{19}$$

In this expression, $\frac{\partial \mathbf{h}_2^{-1}}{\partial \mathbf{x}}(\mathbf{x}_b)$, *is the generalized inverse, i.e.*

$$\frac{\partial \mathbf{h}_2^{-1}}{\partial \mathbf{x}}(\mathbf{x}_b) = \left(\left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{x}}(\mathbf{b}) \right)^T \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{x}}(\mathbf{b}) \right) \right)^{-1} \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{x}}(\mathbf{b}) \right)^T. \tag{20}$$

Proof. The computation will be based on the composition of Figure 7. □

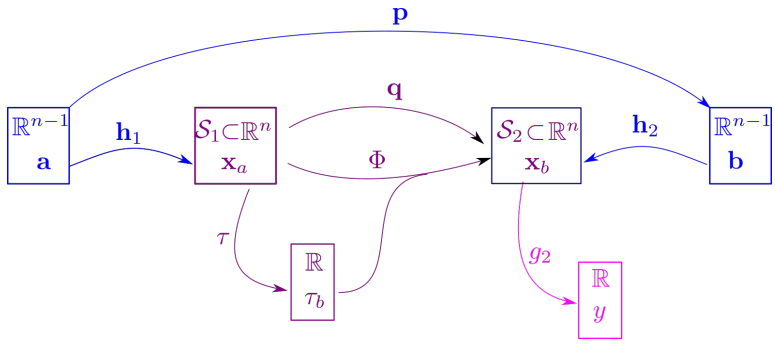


Figure 7: Composition of functions used to compute the Jacobian of the Poincaré map

Define

$$\begin{array}{rcl} y & = & g_2(\mathbf{x}_b) \\ \mathbf{x}_b & = & \mathbf{\Phi}(\mathbf{x}_a, \tau_b) \\ \tau_b & = & \tau(\mathbf{x}_a). \end{array} \tag{21}$$

We have

$$\begin{array}{rcl} dy & = & \frac{\partial g_2}{\partial \mathbf{x}}(\mathbf{x}_b)\cdot d\mathbf{x}_b \\ d\mathbf{x}_b & = & \frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}}(\mathbf{x}_a,\tau_b)\cdot d\mathbf{x}_a + \underbrace{\frac{\partial \mathbf{\Phi}}{\partial t}(\mathbf{x}_a,\tau_b)}_{\mathbf{f}(\mathbf{x}_b)}\cdot d\tau_b \\ d\tau_b & = & \frac{\partial \tau}{\partial \mathbf{x}}(\mathbf{x}_a)\cdot d\mathbf{x}_a. \end{array} \tag{22}$$

Thus

$$\begin{array}{rcl} dy & = & \frac{\partial g_2}{\partial \mathbf{x}}(\mathbf{x}_b)\cdot d\mathbf{x}_b \\ & = & \frac{\partial g_2}{\partial \mathbf{x}}(\mathbf{x}_b)\cdot \left(\frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}}(\mathbf{x}_a,\tau_b)\cdot d\mathbf{x}_a + \mathbf{f}(\mathbf{x}_b)\cdot d\tau_b\right) \\ & = & \frac{\partial g_2}{\partial \mathbf{x}}(\mathbf{x}_b)\cdot \left(\frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}}(\mathbf{x}_a,\tau_b)\cdot d\mathbf{x}_a + \mathbf{f}(\mathbf{x}_b)\cdot \frac{\partial \tau}{\partial \mathbf{x}}(\mathbf{x}_a)\cdot d\mathbf{x}_a\right). \end{array} \tag{23}$$

Since $dy = 0$, we get

$$\frac{\partial g_2}{\partial \mathbf{x}}(\mathbf{x}_b)\cdot \mathbf{f}(\mathbf{x}_b)\cdot \frac{\partial \tau}{\partial \mathbf{x}}(\mathbf{x}_a) = -\frac{\partial g_2}{\partial \mathbf{x}}(\mathbf{x}_b)\cdot \frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}}(\mathbf{x}_a,\tau_b), \tag{24}$$

i.e.

$$\frac{\partial \tau}{\partial \mathbf{x}}(\mathbf{x}_a) = -\frac{1}{\frac{\partial g_2}{\partial \mathbf{x}}(\mathbf{x}_b)\cdot \mathbf{f}(\mathbf{x}_b)}\cdot \frac{\partial g_2}{\partial \mathbf{x}}(\mathbf{x}_b)\cdot \frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}}(\mathbf{x}_a,\tau_b), \tag{25}$$

which corresponds to (iii).

As a consequence

$$\begin{array}{rcl} d\mathbf{x}_b & = & \frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}}(\mathbf{x}_a,\tau_b)\cdot d\mathbf{x}_a + \mathbf{f}(\mathbf{x}_b)\cdot d\tau_b \\ & = & \frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}}(\mathbf{x}_a,\tau_b)\cdot d\mathbf{x}_a + \mathbf{f}(\mathbf{x}_b)\cdot \frac{\partial \tau}{\partial \mathbf{x}}(\mathbf{x}_a)\cdot d\mathbf{x}_a, \end{array} \tag{26}$$

i.e.,

$$\frac{\partial \mathbf{q}}{\partial \mathbf{x}}(\mathbf{x}_a) = \frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}}(\mathbf{x}_a,\tau(\mathbf{x}_a)) + \mathbf{f}(\mathbf{x}_b)\cdot \frac{\partial \tau}{\partial \mathbf{x}}(\mathbf{x}_a), \tag{27}$$

which corresponds to (ii).

The expression for $\mathbf{J_p}(\mathbf{a})$ is directly obtained from the chain rule. ■

Remark 1. The function $\mathbf{h}_2 : \mathcal{A}_2 \mapsto \mathcal{S}$ is a diffeomorphism and we have an expression for it. For instance, it could be

$$\mathbf{h}_2(a_1,a_2) = \begin{pmatrix} a_1+a_2 \\ a_1-a_2 \\ a_1 \end{pmatrix} \tag{28}$$

if we choose $\mathbf{h}_2$ linear. We have $\mathbf{h}_2(\mathbb{R}^2) = \mathcal{S}_2$ which is a two-dimensional plane of $\mathbb{R}^3$. To apply the chain rule, we need the Jacobian matrix for $\mathbf{h}_2^{-1}$. Several expressions exist for it. One of them is the generalized inverse given here by

$$\mathbf{h}_2^{-1} = \begin{pmatrix} \frac{1}{3}(x_1+x_2+x_3) \\ \frac{1}{2}(x_1-x_2) \end{pmatrix}. \tag{29}$$

Indeed

$$\mathbf{h}_2^{-1} \circ \mathbf{h}_2(a_1, a_2) = \begin{pmatrix} \frac{1}{3}(a_1 + a_2 + a_1 - a_2 + a_1) \\ \frac{1}{2}(a_1 + a_2 - (a_1 - a_2)) \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}. \tag{30}$$

Since we have chosen $\mathbf{h}_2$ linear, the function and its Jacobian are similar. The goal of this remark is to explain why we need the generalized inverse (20) whereas $\mathbf{h}_2$ is a diffeomorphism: it is due to the fact that $\mathbf{h}_2$ needs to be represented as a function from $\mathbb{R}^{n-1}$ to $\mathbb{R}^n$.

2.7 Interval extension of its Jacobian

To get an interval extension of the Jacobian matrix $\mathbf{J_p}$ of the Poincaré map, we integrate the variational equation using an interval integration scheme such as the Lohner method [20]. We get a tube $[\mathbf{J}](t)$ and we select the smallest interval matrix which encloses the *monodromy matrix* $[\mathbf{J}]([\tau_b])$, where $[\tau_b]$ is the time interval computed in Section 2.4. The following algorithm computes the Jacobian matrix $\mathbf{J_p}$ for $\mathbf{p}$. Note that this algorithm is not new and can be seen as a simplified version of existing algorithms, see *e.g.,* [15, 35, 36].

Algorithm IntervalPoincaréJacobian

Input: $\mathbf{f}, [\mathbf{a}]$

Output: $[\mathbf{J_p}]$

1: $[\mathbf{x}_a] = [\mathbf{h}_1]([\mathbf{a}])$

2: Compute the tubes $[\mathbf{x}](t) = [\mathbf{\Phi}]([\mathbf{x}_a], t)$ and $[\mathbf{J}](t) = [\frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}}]([\mathbf{x}_a], t)$

3: From $g_2([\mathbf{x}](t))$, select the Poincaré time interval $[\tau_b]$ (see Section 2.4)

4: $[\mathbf{x}_b] = [\mathbf{x}]([\tau_b])$

5: $[\mathbf{J}_m] = [\mathbf{J}]([\tau_b])$ *(monodromy matrix)*

6: $[\mathbf{J}_q] = \left(\mathbf{I} - \frac{[\mathbf{f}]([\mathbf{x}_b]) \cdot \left([\frac{\partial g_2}{\partial \mathbf{x}}]([\mathbf{x}_b])\right)}{[\mathbf{J}_{g_2}] \cdot [\mathbf{f}]([\mathbf{x}_b])}\right) \cdot [\mathbf{J}_m]$

7: $[\mathbf{b}] = [\mathbf{h}_2^{-1}]([\mathbf{x}_b])$

8: $[\mathbf{H}_2] = [\frac{\partial \mathbf{h}_2}{\partial \mathbf{b}}]([\mathbf{b}])$

9: $[\mathbf{J_p}] = \left(\left([\mathbf{H}_2^{\mathrm{T}}] \cdot [\mathbf{H}_2]\right)^{-1} [\mathbf{H}_2^{\mathrm{T}}]\right) \cdot [\mathbf{J}_q] \cdot [\frac{\partial \mathbf{h}_1}{\partial \mathbf{a}}]([\mathbf{a}])$

2.8 Example

We choose a very simple example to illustrate the principle of the procedure IntervalPoincaréJacobian. We have chosen this example for the following reasons:

- It is related to the application that will be considered in Section 3.

- It can be executed analytically by hand by the reader for a better understanding of the approach.
- The example will allow us to produce a 3D figure which illustrates clearly the principle of our method.

Consider the system

$$\left\{\begin{array}{lcl} \dot{x}_1 &=& 1 \\ \dot{x}_2 &=& \sin x_3 \\ \dot{x}_3 &=& 1. \end{array}\right. \tag{31}$$

We assume that we have two surfaces

$$\begin{array}{lcl} \mathcal{S}_1 &=& \{\mathbf{x} \,|\, x_1 = 0\} \\ \mathcal{S}_2 &=& \{\mathbf{x} \,|\, x_2 = 0\}. \end{array} \tag{32}$$

We fix the coordinate frames of these surfaces by choosing the origins $\mathbf{o}_1 = (0, 0, \frac{\pi}{2})$, $\mathbf{o}_2 = (\pi, 0, \frac{3\pi}{2})$ and the basis $\mathbf{i}_1 = (0, 1, 0)$, $\mathbf{j}_1 = (0, 0, 1)$ for $\mathcal{S}_1$. The basis for $\mathcal{S}_2$ is chosen as $\mathbf{i}_2 = (1, 0, 0)$, $\mathbf{j}_2 = (0, 0, 1)$. Thus, the charts are

$$\mathbf{h}_1(\mathbf{a}) = \begin{pmatrix} 0 \\ a_1 \\ a_2 \end{pmatrix} + \mathbf{o}_1 = \begin{pmatrix} 0 \\ a_1 \\ a_2 + \frac{\pi}{2} \end{pmatrix}, \tag{33}$$

and

$$\mathbf{h}_2(\mathbf{b}) = \begin{pmatrix} b_1 \\ 0 \\ b_2 \end{pmatrix} + \mathbf{o}_2 = \begin{pmatrix} b_1 + \pi \\ 0 \\ b_2 + \frac{3\pi}{2} \end{pmatrix}. \tag{34}$$

Take $[\mathbf{a}] = [-0.1, 0.1] \times [-0.1, 0.1]$.
Step 1. We have

$$[\mathbf{x}_a] = [\mathbf{h}_1]([\mathbf{a}]) = \begin{pmatrix} 0 \\ [-0.1, 0.1] \\ [-0.1 + \frac{\pi}{2}, 0.1 + \frac{\pi}{2}] \end{pmatrix}. \tag{35}$$

Step 2. We need to consider the variational equation:

$$\dot{\mathbf{J}} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \cdot \mathbf{J}, \tag{36}$$

i.e.,

$$\begin{pmatrix} \dot{J}_{11} & \dot{J}_{12} & \dot{J}_{13} \\ \dot{J}_{21} & \dot{J}_{22} & \dot{J}_{23} \\ \dot{J}_{31} & \dot{J}_{32} & \dot{J}_{33} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \cos x_3 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{pmatrix}. \tag{37}$$

Using an interval integration for both (31) and (37), for an initial vector $[\mathbf{x}_a]$ we get a tube for $[\mathbf{x}](t) = [\mathbf{\Phi}]([\mathbf{x}_a], t)$ and a tube for $[\mathbf{J}](t) = [\frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}}]([\mathbf{x}_a], t)$.

Step 3. Since $g_2(\mathbf{x}) = x_2$, we get the Poincaré interval $[\tau_b]$ from the second component of $[\mathbf{x}](t)$. We get

$$\tau_b \in [\tau_b] = [2.82, 3.47]. \tag{38}$$

Step 4. From the tube $[\mathbf{x}](t)$, we extract $[\mathbf{x}_b] = [\mathbf{x}]([\tau_b])$. We get

$$[\mathbf{x}_b] = [2.82, 3.47] \times [-0.63, 0.63] \times [4.29, 5.14]. \tag{39}$$

Step 5,**6**. We get

$$[\mathbf{J}_q] = \begin{pmatrix} 1 & [1, 1.1] & [-2.41, -1.60] \\ [-0.01, 0.01] & [-0.1, 0.09] & [-0.64, 0.7] \\ [-0.01, 0.01] & [1, 1.1] & [-1.41, -0.69] \end{pmatrix}. \tag{40}$$

Step 7. We get (see Figure 8)

$$[\mathbf{b}] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \left([\mathbf{x}_b] - \begin{pmatrix} \pi \\ 0 \\ \frac{3\pi}{2} \end{pmatrix} \right). \tag{41}$$

Step 8. We get the degenerate matrix

$$[\mathbf{H}_2] = [\frac{\partial \mathbf{h}_2}{\partial \mathbf{x}}]([\mathbf{b}])) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{42}$$

Step 9. We get

$$[\mathbf{J_p}] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot [\mathbf{J}_q] \cdot \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} [0.99, 1.1] & [-2.41, -1.60] \\ [0.99, 1.1], & [-1.41, -0.69] \end{pmatrix}. \tag{43}$$

3 Rolling navigation

3.1 Principle

We consider the robot moving on a plane described by the Dubins car model [10]

$$\begin{cases} \dot{x}_1 &= \cos x_3 \\ \dot{x}_2 &= \sin x_3 \\ \dot{x}_3 &= u, \end{cases} \tag{44}$$

where (x_1, x_2) is the position of the robot, x_3 is its heading and u is the input. The robot has no possibility to measure its state, neither its position nor its heading. It is only able to measure a function $\varphi(x_1, x_2)$ of its position such as a temperature or

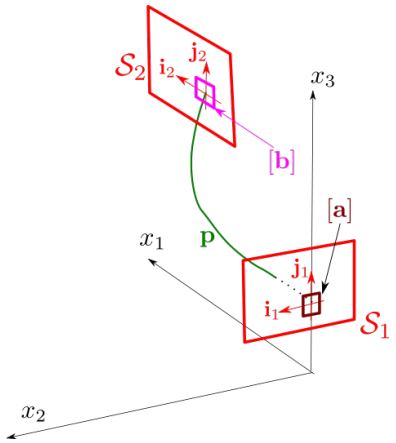


Figure 8: Example of Poincaré interval extension

an altitude. We want that the robot moves along the *wanted* curve $\varphi(x_1, x_2) = 0$. For this, we suggest to use the Trinity pattern proposed in [34] which yields a *rolling* behavior for the motion. The stability of the resulting navigation has been shown experimentally in [34] with an autonomous plane turning around a cloud with an unknown shape.

Now, to our knowledge, no theoretical analysis has been provided in the literature. The principle of the rolling navigation is to alternate between a circle of radius ρ_0 when $\varphi < 0$ and a circle of radius ρ_1 when $\varphi > 0$, as illustrated by Figure 9. The left figure illustrates the ideal situation where the robot starts on the wanted curve $\varphi = 0$ (which is approximated by a line) with an incident angle of $\frac{\pi}{2}$. The robot follows the circle of radius ρ_0 until $\varphi = 0$, taking $u = \frac{1}{\rho_0}$. With a counter, the robot measures the elapsed time c_0. We should have $c_0 = \rho_0\pi$. Then the robot follows the circle of radius ρ_1 for a time c_1 in order to be on the wanted line again. For k_1, k_2 in $\mathbb{N}$, we should have

$$k_1\frac{c_1}{\rho_1} + k_2\frac{c_0}{\rho_0} = (k_1 + k_2)\pi, \tag{45}$$

i.e.,

$$c_1 = \rho_1\left(\pi + \frac{k_2}{k_1}\left(\pi - \frac{c_0}{\rho_0}\right)\right). \tag{46}$$

Take for instance $k_1 = 2$, $k_2 = 1$. We get

$$c_1 \quad = \quad \rho_1\frac{3\pi - \frac{c_0}{\rho_0}}{2} = \rho_1\pi. \tag{47}$$

If we have no uncertainties, the robot will be on the wanted line with an incidence angle of $\frac{\pi}{2}$.

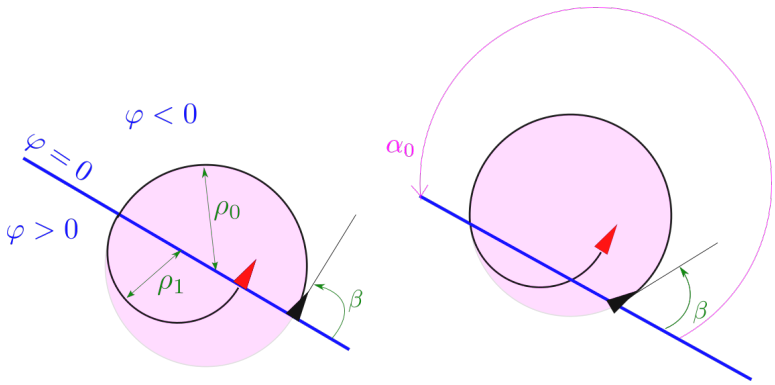


Figure 9: Left: The incidence angle $\beta = \frac{\pi}{2}$ is the right. Right: $\beta \neq \frac{\pi}{2}$ should be compensated

3.2 Stability along the path

The question we need to study now, is the stability along the path $\varphi = 0$ for the chosen k_1, k_2. Consider the case where $\beta \neq \frac{\pi}{2}$ and assume that we are on the wanted line. The robot follows the circle of radius ρ_0 until $\varphi = 0$, taking $u = \frac{1}{\rho_0}$. It measures an elapsed time of $c_0 = \rho_0 \alpha_0$, where α_0 is the corresponding angle and then follows the circle of radius ρ_1 for a time c_1 given by (47). We understand that we are not anymore on the wanted line and proving the stability is not trivial.

Take for simplicity $\rho_0 = 1, \rho_1 = \frac{1}{2}$.

Figure 10 shows a block diagram with the Dubins car and the controller. The controller has a single input corresponding to φ. It has two state variables: $q \in \{0, 1\}$ and the counter $c \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. It generates the control u.

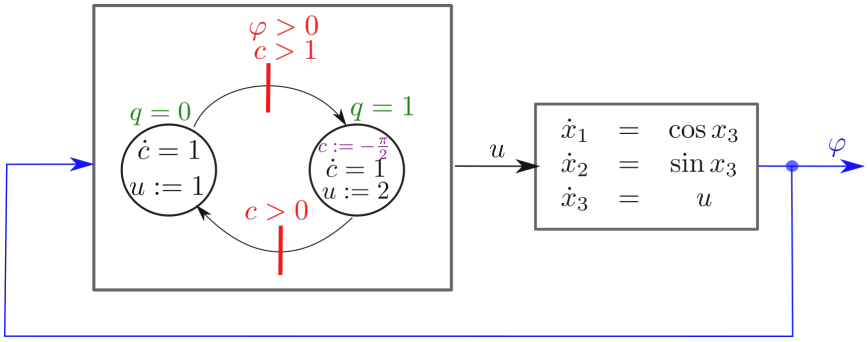


Figure 10: The controller is an automaton which controls our robot

The hybrid automaton representing our controlled system is given by Figure 11.

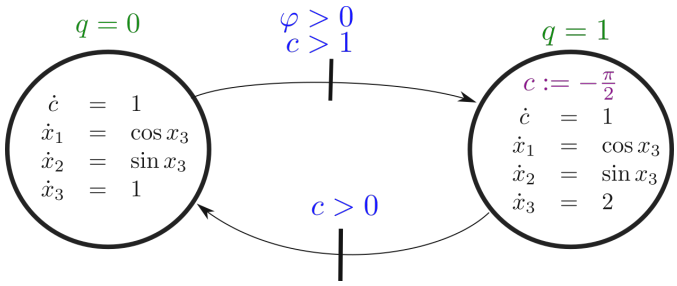


Figure 11: The corresponding trajectory is expected to roll on the curve $\varphi = 0$ in a stable way

3.3 Stability of a periodic orbit

Stability along a path can be simplified by a stability along a periodic orbit in the state space, by taking into account the symmetries by translation and by rotation of the problem. We consider a linear approximation of φ and we change the coordinate frame so that $\varphi > 0$ translates into $x_2 < 0$. This is illustrated by Figure 12.

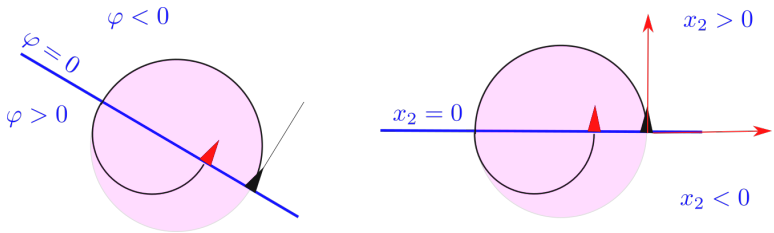


Figure 12: Left: The function φ is assumed to be linear. Right: simplification after a change of the coordinate frame

We want to prove the stability with respect to (q, c, x_2, x_3) at the point $(0, 0, 0, \frac{\pi}{2})$. The corresponding hybrid automaton is depicted in Figure 13. The state variable x_1 has been removed since its stability is not of interest.

4 Proving the stability of rolling navigation

In this section, we propose to use analytical expressions of the Poincaré maps and their Jacobian to have a better understanding of how they are computed. Later, we will show that we do not need any analytical expression to prove the stability.

For the sake of clarity, we added an intermediate state $q = \frac{1}{2}$, as illustrated by Figure 14. This state, called the *jump*, is fleeting, *i.e.*, the state stays inside the

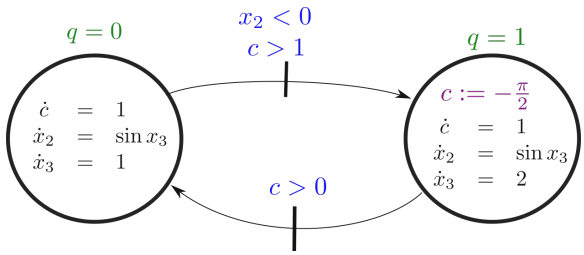


Figure 13: The periodic orbit generated by this automaton is stable if we start at $(q, c, x_2, x_3) = (0, 0, 0, \frac{\pi}{2})$

jump state for 0 sec, or equivalently, as soon as it is inside $q = \frac{1}{2}$, it jumps to $q = 1$. In the jump, we added $x_3 := x_3 - 2\pi$ which is a non transformation, since x_3 is an angle. Now, this transformation allows us to have a bounded x_3 and to reason in the Cartesian line for x_3 instead of the trigonometric circle. Otherwise, the angle x_3 would increase by 2π at each lap of the hybrid automaton.

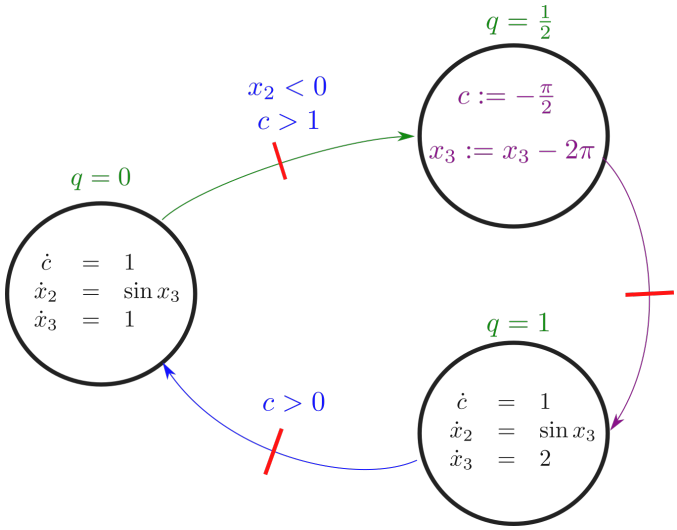


Figure 14: The trajectory generated by this automaton is stable at $(q, c, x_2, x_3) = (0, 0, 0, \frac{\pi}{2})$

4.1 Periodic orbit

If we simulate the system, starting from $(q, c, x_2, x_3) = (0, 0, 0, \frac{\pi}{2})$, we get the periodic orbit depicted in Figure 15 in the (c, x_2, x_3)-space. The red transitions of Figure 14 become the three red two-dimensional Poincaré sections of Figure 15. We switch from one surface to another using the partial Poincaré maps $\mathbf{p}_0, \mathbf{p}_{\frac{1}{2}}, \mathbf{p}_1$.

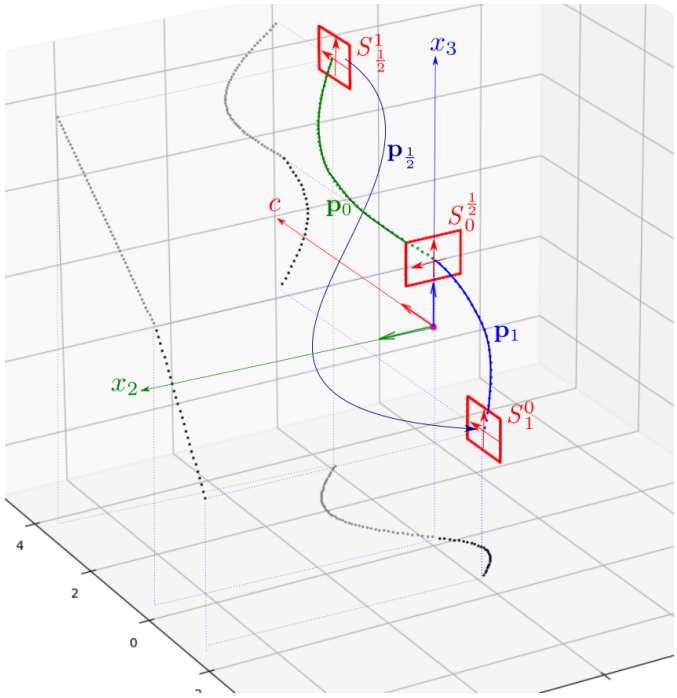


Figure 15: Periodic orbit in the (c, x_2, x_3)-space. The gray curves correspond to the three projections of the trajectory on each of the three canonical vertical planes

4.2 Equilibrium

Proposition. Assume that at time $t = 0$, we are at the state

$$(q, c, x_2, x_3) = (0, 0, 0, \frac{\pi}{2}). \tag{48}$$

Then at time $t = \frac{3\pi}{2}$, we will come back to the same state.

Proof. Let us start at $t = 0$. We have

$$\begin{array}{rcl} c(t) & = & t \\ x_2(t) & = & x_2(0) + \int_0^t \sin(x_3(\tau)) \cdot d\tau = \int_0^t \sin(\frac{\pi}{2} + \tau) \cdot d\tau \\ & = & \int_0^t \cos(\tau) \cdot d\tau = \sin(t) \\ x_3(t) & = & \frac{\pi}{2} + t. \end{array} \tag{49}$$

When $t = \pi$, we have $x_2(t) = 0$ and we switch to $q = \frac{1}{2}$. The state is now

$$(q, c, x_2, x_3) = (\frac{1}{2}, \pi, 0, \frac{3\pi}{2}). \tag{50}$$

We immediately jump to $(1, -\frac{\pi}{2}, 0, -\frac{\pi}{2})$.

As long as we stay with $q = 1$, we have

$$\begin{array}{rcl} c(t) & = & -\frac{\pi}{2} + (t - \pi) \\ x_2(t) & = & x_2(\pi) + \int_\pi^t \sin(x_3(\tau)) \cdot d\tau \\ & = & -\int_\pi^t \sin(\frac{\pi}{2} - 2\tau) \cdot d\tau = -\int_\pi^t \cos(2\tau) \cdot d\tau \\ & = & -\left[\frac{1}{2}\sin(2\tau)\right]_\pi^t = -\frac{1}{2} \cdot \sin(2t) \\ x_3(t) & = & x_3(\pi) + \int_\pi^t 2 \cdot d\tau = -\frac{\pi}{2} + 2(t - \pi) = 2t - \frac{5\pi}{2}. \end{array} \tag{51}$$

When $t = \frac{3\pi}{2}$, we have $c(t) = 0$ and switch back to $q = 0$. The state it now

$$\begin{array}{rcl} (q, c, x_2, x_3) & = & (0, 0, -\frac{1}{2} \cdot \sin(2 \cdot \frac{3\pi}{2}), 2 \cdot \frac{3\pi}{2} - \frac{5\pi}{2}) \\ & = & (0, 0, 0, \frac{\pi}{2}). \end{array} \tag{52}$$

We thus came back to the initial state.

4.3 Expression for $\mathbf{p}_0$

Take $t = 0$, and assume that we are at the state

$$(q, c, x_2, x_3) = (0, 0, \tilde{x}_2, \frac{\pi}{2} + \tilde{x}_3). \tag{53}$$

It means that we are on the Poincaré surface $\mathcal{S}_0^{\frac{1}{2}}$, at the coordinates $\tilde{x}_2, \tilde{x}_3$.

As long as we satisfy $q = 0$, we have

$$\begin{array}{rcl} c(t) & = & t \\ x_2(t) & = & \tilde{x}_2 + \int_0^t \sin(x_3(\tau)) \cdot d\tau = \tilde{x}_2 + \int_0^t \sin(\frac{\pi}{2} + \tau + \tilde{x}_3) \cdot d\tau \\ & = & \tilde{x}_2 + \int_0^t \cos(\tau + \tilde{x}_3) \cdot d\tau = \tilde{x}_2 + [\sin(\tau + \tilde{x}_3)]_0^t \\ & = & \tilde{x}_2 + \sin(t + \tilde{x}_3) - \sin(\tilde{x}_3) \\ x_3(t) & = & \frac{\pi}{2} + \tilde{x}_3 + t. \end{array} \tag{54}$$

We define

$$\begin{array}{rcl} t_1(\tilde{x}_2, \tilde{x}_3) & = & \min\{t > 0 \,|\, \tilde{x}_2 + \sin(t + \tilde{x}_3) - \sin(\tilde{x}_3) = 0\} \\ & = & \min\{t > 0 \,|\, \sin(t + \tilde{x}_3) = \sin(\tilde{x}_3) - \tilde{x}_2\} \\ & = & \pi - \arcsin(\sin(\tilde{x}_3) - \tilde{x}_2) - \tilde{x}_3. \end{array} \tag{55}$$

We thus get the first partial Poincaré map

$$\begin{array}{rcl} \mathbf{p}_0 \begin{pmatrix} \tilde{x}_2 \\ \tilde{x}_3 \end{pmatrix} & = & \begin{pmatrix} t_1(\tilde{x}_2, \tilde{x}_3) \\ \frac{\pi}{2} + \tilde{x}_3 + t_1(\tilde{x}_2, \tilde{x}_3) \end{pmatrix} - \begin{pmatrix} \pi \\ \frac{3\pi}{2} \end{pmatrix} \\ & = & \begin{pmatrix} -\arcsin(\sin(\tilde{x}_3) - \tilde{x}_2) - \tilde{x}_3 \\ -\arcsin(\sin(\tilde{x}_3) - \tilde{x}_2) \end{pmatrix}. \end{array} \tag{56}$$

A first order approximation of this function is

$$\begin{aligned}\mathbf{p}_0\begin{pmatrix} dx_2 \\ dx_3 \end{pmatrix} &= \begin{pmatrix} -\arcsin(\sin(dx_3)-dx_2)-dx_3 \\ -\arcsin(\sin(dx_3)-dx_2) \end{pmatrix} \\ &= \begin{pmatrix} -2dx_3+dx_2 \\ -dx_3+dx_2 \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} 1 & -2 \\ 1 & -1 \end{pmatrix}}_{=\mathbf{J}_0}\begin{pmatrix} dx_2 \\ dx_3 \end{pmatrix}. \end{aligned} \tag{57}$$

These results are consistent with those obtained in Section 2.8.

4.4 Expression for $\mathbf{p}_{\frac{1}{2}}$

The jump is the affine map defined by

$$\mathbf{p}_{\frac{1}{2}}\begin{pmatrix} \tilde{c} \\ \tilde{x}_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \tilde{c} \\ \tilde{x}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ \tilde{x}_3 \end{pmatrix}. \tag{58}$$

Thus, we get

$$\mathbf{p}_{\frac{1}{2}} \circ \mathbf{p}_0\begin{pmatrix} [-0.1,0.1] \\ [-0.1,0.1] \end{pmatrix} \subset \begin{pmatrix} 0 \\ [-0.43,0.43] \end{pmatrix} \tag{59}$$

and

$$\mathbf{J}_{\mathbf{p}_{\frac{1}{2}}} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{60}$$

4.5 Expression for $\mathbf{p}_1$

Take $t = 0$, and assume that we are at the state

$$(q, c, x_2, x_3) = (1, -\frac{\pi}{2} + \tilde{c}, 0, -\frac{\pi}{2} + \tilde{x}_3). \tag{61}$$

As long as we satisfy $q = 1$, we have

$$\begin{aligned} c(t) &= -\tfrac{\pi}{2} + \tilde{c} + t \\ x_2(t) &= 0 + \textstyle\int_0^t \sin(x_3(\tau)) \cdot d\tau = \int_0^t -\sin(\tfrac{\pi}{2} - 2\tau - \tilde{x}_3) \cdot d\tau \\ &= -\textstyle\int_0^t \cos(2\tau + \tilde{x}_3) \cdot d\tau \\ &= -\left[\tfrac{1}{2}\sin(2\tau + \tilde{x}_3)\right]_0^t = \tfrac{1}{2}\sin(\tilde{x}_3) - \tfrac{1}{2}\sin(2t + \tilde{x}_3) \\ x_3(t) &= -\tfrac{\pi}{2} + \tilde{x}_3 + 2t. \end{aligned} \tag{62}$$

We define

$$t_2(\tilde{c}, \tilde{x}_3) = \min\left\{t > 0 \,|\, -\frac{\pi}{2} + \tilde{c} + t = 0\right\} = \frac{\pi}{2} - \tilde{c}. \tag{63}$$

We have

$$\begin{aligned}
\mathbf{p}_1 \begin{pmatrix} \tilde{c} \\ \tilde{x}_3 \end{pmatrix} &= \begin{pmatrix} \frac{1}{2}\sin(\tilde{x}_3) - \frac{1}{2}\sin(2t_2(\tilde{c},\tilde{x}_3) + \tilde{x}_3) \\ -\frac{\pi}{2} + \tilde{x}_3 + 2t_2(\tilde{c},\tilde{x}_3) \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{\pi}{2} \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{2}\sin(\tilde{x}_3) - \frac{1}{2}\sin\left(2\left(\frac{\pi}{2} - \tilde{c}\right) + \tilde{x}_3\right) \\ -\frac{\pi}{2} + \tilde{x}_3 + 2\left(\frac{\pi}{2} - \tilde{c}\right) - \frac{\pi}{2} \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{2}\sin(\tilde{x}_3) - \frac{1}{2}\sin(\pi - 2\tilde{c} + \tilde{x}_3) \\ \tilde{x}_3 - 2\tilde{c} \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{2}\sin(\tilde{x}_3) - \frac{1}{2}\sin(2\tilde{c} - \tilde{x}_3) \\ \tilde{x}_3 - 2\tilde{c} \end{pmatrix}.
\end{aligned} \tag{64}$$

A first order approximation of $\mathbf{p}_1$ is

$$\begin{aligned}
\mathbf{p}_1 \begin{pmatrix} dc \\ dx_3 \end{pmatrix} &= \begin{pmatrix} \frac{1}{2}\sin(dx_3) - \frac{1}{2}\sin(2dc - dx_3) \\ dx_3 - 2dc \end{pmatrix} \\
&= \begin{pmatrix} -dc + dx_3 \\ dx_3 - 2dc \end{pmatrix} \\
&= \underbrace{\begin{pmatrix} -1 & 1 \\ -2 & 1 \end{pmatrix}}_{=\mathbf{J}_1} \begin{pmatrix} dc \\ dx_3 \end{pmatrix}.
\end{aligned} \tag{65}$$

4.6 Poincaré map

We define

$$\mathbf{p} \begin{pmatrix} \tilde{x}_2 \\ \tilde{x}_3 \end{pmatrix} = \mathbf{p}_1 \circ \mathbf{p}_{\frac{1}{2}} \circ \mathbf{p}_0 \begin{pmatrix} \tilde{x}_2 \\ \tilde{x}_3 \end{pmatrix}. \tag{66}$$

Since

$$\mathbf{p}_0 \begin{pmatrix} \tilde{x}_2 \\ \tilde{x}_3 \end{pmatrix} \mapsto \begin{pmatrix} -\arcsin(\sin(\tilde{x}_3) - \tilde{x}_2) - \tilde{x}_3 \\ -\arcsin(\sin(\tilde{x}_3) - \tilde{x}_2) \end{pmatrix} \tag{67}$$

$$\mathbf{p}_{\frac{1}{2}} \begin{pmatrix} \tilde{c} \\ \tilde{x}_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{c} \\ \tilde{x}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ \tilde{x}_3 \end{pmatrix} \tag{68}$$

$$\mathbf{p}_1 \begin{pmatrix} \tilde{c} \\ \tilde{x}_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\sin(\tilde{x}_3) - \frac{1}{2}\sin(2\tilde{c} - \tilde{x}_3) \\ \tilde{x}_3 - 2\tilde{c} \end{pmatrix}, \tag{69}$$

we get

$$\mathbf{p} \begin{pmatrix} \tilde{x}_2 \\ \tilde{x}_3 \end{pmatrix} = \begin{pmatrix} \tilde{x}_2 - \sin(\tilde{x}_3) \\ \arcsin(\tilde{x}_2 - \sin(\tilde{x}_3)) \end{pmatrix}. \tag{70}$$

The Jacobian of $\mathbf{p}$ at $\mathbf{0}$ is

$$\mathbf{J_p}(\mathbf{0}) = \mathbf{J}_1 \cdot \mathbf{J}_{\frac{1}{2}} \cdot \mathbf{J}_0 = \begin{pmatrix} -1 & 1 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -2 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}, \quad (71)$$

which is stable (all roots are in the unit disk). As a consequence, the periodic orbit is stable.

5 Basin of attraction

In this section, we want to find a subset of the state space of all states which will converge to the periodic orbit. It is sufficient to limit our analysis to a Poincaré section, say S_0^1. More precisely, we will find a centered box $\mathbb{X}_0$ inside S_0^1 such that $\mathbf{p}(\mathbb{X}_0) \subset \mathbb{X}_0$ in this case, $\mathbb{X}_0$ is positive invariant. Unfortunately, in our situation (which is not atypical), such a box does not exist and we can only find $k > 1$ such that $\mathbf{p}^k(\mathbb{X}_0) \subset \mathbb{X}_0$. This corresponds to the *periodic invariance* studied by Lee and Kouvaritakis in [18].

5.1 Find a periodic positive invariant box

To check the stability we take a small box containing 0, for instance

$$\mathbb{X}_0 = [-0.1, 0.1] \times [-0.1, 0.1],$$

which corresponds to the red box in Figure 16. If we compute the smallest box which contains $\mathbb{X}_0$ we find the blue box, which means that $\mathbf{p}(\mathbb{X}_0) \not\subset \mathbb{X}_0$. Now, we also get $\mathbf{p}^2(\mathbb{X}_0) \subset \mathbb{X}_0$ and $\mathbf{p}^3(\mathbb{X}_0) \subset \mathbb{X}_0$. We conclude that $\mathbb{X}_0$ is periodic positive invariant.

5.2 Find an asymptotically stable box

We now want to show that all initial state inside $\mathbb{X}_0$ will converge to 0. We use the centered form [22] for stability [8, 28]. For this, we follow the procedure given by relation (7). For $k = 1$, we do not get the enclosure. For $k = 2$, we get (see Equation (9)):

$$\left([\mathbf{J_p}]\left([\mathbf{p}]\begin{pmatrix}[\tilde{x}_2]\\[\tilde{x}_3]\end{pmatrix}\right)\right) \cdot \left([\mathbf{J_p}]\begin{pmatrix}[\tilde{x}_2]\\[\tilde{x}_3]\end{pmatrix}\right) \cdot \begin{pmatrix}[\tilde{x}_2]\\[\tilde{x}_3]\end{pmatrix} \subset \begin{pmatrix}[\tilde{x}_2]\\[\tilde{x}_3]\end{pmatrix}. \quad (72)$$

For the Poincaré map and its Jacobian, we took:

$$\mathbf{p}\begin{pmatrix}\tilde{x}_2\\\tilde{x}_3\end{pmatrix} = \begin{pmatrix}\tilde{x}_2 - \sin(\tilde{x}_3)\\ \arcsin(\tilde{x}_2 - \sin(\tilde{x}_3))\end{pmatrix} \quad (73)$$

and

$$\mathbf{J_p}\begin{pmatrix}\tilde{x}_2\\\tilde{x}_3\end{pmatrix} = \begin{pmatrix} 1 & -\cos(\tilde{x}_3) \\ \frac{1}{\sqrt{1-(\tilde{x}_2-\sin(\tilde{x}_3))^2}} & \frac{-\cos(\tilde{x}_3)}{\sqrt{1-(\tilde{x}_2-\sin(\tilde{x}_3))^2}} \end{pmatrix}. \quad (74)$$

Following [8] we get that all trajectories initialized in $\mathbb{X}_0$ will converge to $\mathbf{0}$ and will stay inside $\mathbf{p}(\mathbb{X}_0)$. We can write this property under the form $\mathbf{p}^\infty(\mathbb{X}_0) = \mathbf{0}$.

5.3 Find an asymptotically stable box using an interval integration

In the general case, we do not have any analytical expression for the flow. The procedure has to be applied using an interval integration. We give here all intermediate results related to our test-case (see [7] for more details). We start from

$$[\mathbf{a}] = [-0.01, 0.01]. \tag{75}$$

First lap. We get

$$[\tau_b] = [3.1114, 3.17179]\,, \tag{76}$$

$$[\mathbf{x}_b] = \begin{pmatrix} [3.1114, 3.17179] \\ [-0.0603952, 0.0603829] \\ [4.67219, 4.75259] \end{pmatrix}. \tag{77}$$

The monodromy matrix for $\mathbf{p}_0$ is

$$[\mathbf{J}]([\tau_b]) = \begin{pmatrix} [1,1] & \left[-10^{-10}, 10^{-10}\right] & \left[-10^{-10}, 10^{-10}\right] \\ \left[-10^{-10}, 10^{-10}\right] & [1,1] & [-2.01996, -1.97913] \\ \left[-10^{-10}, 10^{-10}\right] & \left[-10^{-10}, 10^{-10}\right] & [1,1] \end{pmatrix} \tag{78}$$

and the Jacobian matrix for $\mathbf{p}_0$ is

$$[\mathbf{J}_{\mathbf{p}_0}] = \begin{pmatrix} [1, 1.00081] & [-2.02159, -1.97913] \\ [1, 1.00081] & [-1.02159, -0.979133] \end{pmatrix}. \tag{79}$$

We have the jump and then, we switch to $\mathbf{p}_1$. We get

$$[\tau_b] = [1.5708, 1.5709] \tag{80}$$

and

$$[\mathbf{x}_b] = \begin{pmatrix} \left[-10^{-10}, 10^{-10}\right] \\ [-0.0417802, 0.0417865] \\ [1.5306, 1.611] \end{pmatrix}. \tag{81}$$

For the monodromy matrix, we get

$$[\mathbf{J}]([\tau_b]) = \begin{pmatrix} [1,1] & \left[-10^{-10}, 10^{-10}\right] & \left[-10^{-10}, 10^{-10}\right] \\ \left[-10^{-10}, 10^{-10}\right] & [1,1] & [0.958986, 1.03941] \\ \left[-10^{-10}, 10^{-10}\right] & \left[-10^{-10}, 10^{-10}\right] & [1,1] \end{pmatrix}. \tag{82}$$

The Jacodian matrix is

$$[\mathbf{J}_{\mathbf{p}_1}] = \begin{pmatrix} [1, 1.00081] & [-2.02159, -1.97913] \\ [1, 1.00081] & [-1.02159, -0.979133] \end{pmatrix}, \tag{83}$$

and the interval enclosure of the Poincaré map becomes

$$[\mathbf{p}]([\mathbf{a}]) = \begin{pmatrix} [-0.021021, 0.021021] \\ [-0.020224, 0.020224] \end{pmatrix}. \tag{84}$$

Second lap. We perform the same type of computation as for the first lap and we get

$$[\mathbf{p} \circ \mathbf{p}]([\mathbf{a}]) = \begin{pmatrix} [-0.00512577, 0.00512577] \\ [-0.00178467, 0.00178467] \end{pmatrix}. \tag{85}$$

For all other details, see [7].

From these results, we have easily checked that $[\mathbf{J}_\mathbf{p}]([\mathbf{p}][\mathbf{a}]) \cdot [\mathbf{J}_\mathbf{p}]([\mathbf{a}]) \cdot [\mathbf{a}] \subset [\mathbf{a}]$ and we conclude the asymptotic stability.

5.4 Capture basin

We now want to characterize a set larger than $\mathbb{X}_0 = [-0.1, 0.1] \times [-0.1, 0.1]$ for $(\tilde{x}_2, \tilde{x}_3)$ which will converge to $\mathbf{0}$. Such a set is called a *capture basin* [2]. We know from [3, 4] that, since $\mathbb{X}_0$ is a capture basin, $\mathbf{p}^{-k}(\mathbb{X}_0), k \geq 0$ is also a basin. Indeed all points $(\tilde{x}_2, \tilde{x}_3) \in \mathbf{p}^{-k}(\mathbb{X}_0)$ will be such that $\mathbf{p}^k(\tilde{x}_2, \tilde{x}_3) \in \mathbb{X}_0$ and will thus converge to $\mathbf{0}$.

The orange strip in Figure 16 corresponds $\mathbb{X}_1 = \mathbf{p}^{-1}(\mathbb{X}_0)$ and extends from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$, The green set corresponds to $\mathbb{X}_2 = \mathbf{p}^{-2}(\mathbb{X}_0)$.

All the properties we have proven can be summarized by Figure 17. From this graph, we read that $\mathbf{p}^k(\mathbb{X}_2) \subset \mathbb{X}_0$ for $k \in \{2, 4, 5, 6, \dots\}$. But, we have $\mathbf{p}^3(\mathbb{X}_2) \subset [\mathbf{b}]$ which may be outside $\mathbb{X}_0$. Moreover, we have $\mathbf{p}^\infty(\mathbb{X}_2) = \mathbf{0}$. The non inclusion monotonicity of the chain is due to the fact that $\mathbb{X}_0$ is not positive invariant. It is only *periodic* positive invariant.

5.5 Illustration

The stability property has been proven for a linear $\varphi(x_1, x_2)$. In order to illustrate the behavior of our controller for an arbitrary φ, we consider that φ corresponds to the Hippopede of Proclus given by

$$\varphi(x_1, x_2) = 9x_1^2 + x_2^2 - \left(x_1^2 + y_2^2\right)^2. \tag{86}$$

Of course, this equation is not known by our controller which is based on the fact that $\varphi(x_1, x_2)$ is linear. We take for the initial state vector of the robot $\mathbf{x} = (3, 0, 1)$ and for the controller $q = 0$, $c = 0$. The simulation of the controlled robot generates

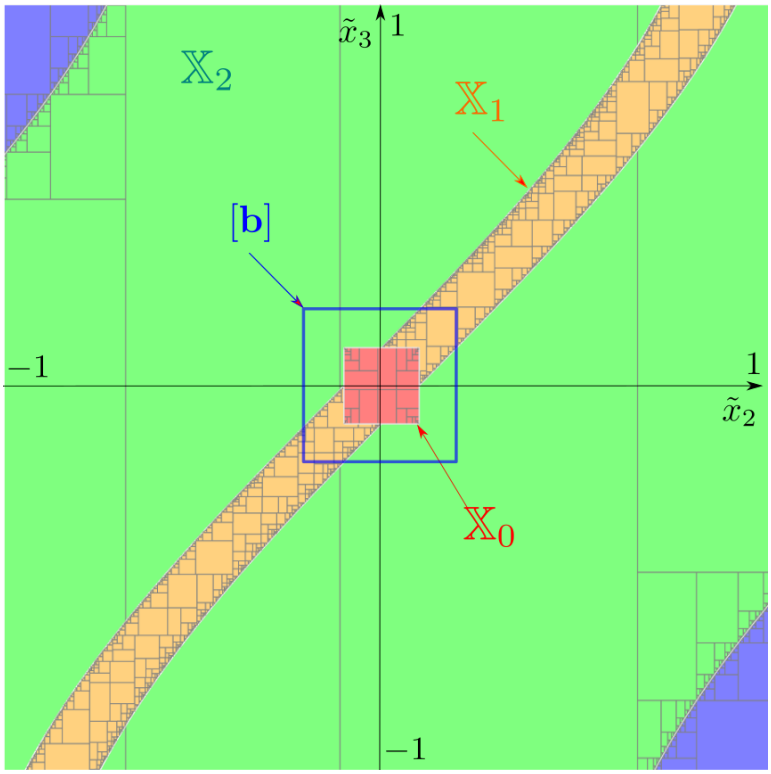


Figure 16: Each point in $\mathbb{X}_2$ (green) will first enter in the orange set $\mathbb{X}_1$, then in the red set $\mathbb{X}_0$. Once in $\mathbb{X}_0$, it will converge to $\mathbf{0}$

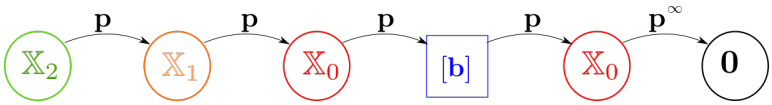


Figure 17: Each point in $\mathbb{X}_2$ (green) will converge to $\mathbf{0}$ and will cross a non-nested chain of sets

Figure 18 and shows that without any compass, measuring only the sign of a scalar unknown function φ depending of the position, we are able to move along the curve $\varphi(x_1, x_2) = 0$ in a stable and robust way.

6 Conclusion

In this paper, we have proposed an interval extension of Poincaré maps to show the stability of hybrid dynamical systems with respect to a periodic orbit. The approach has been illustrated on the rolling navigation. This type of navigation

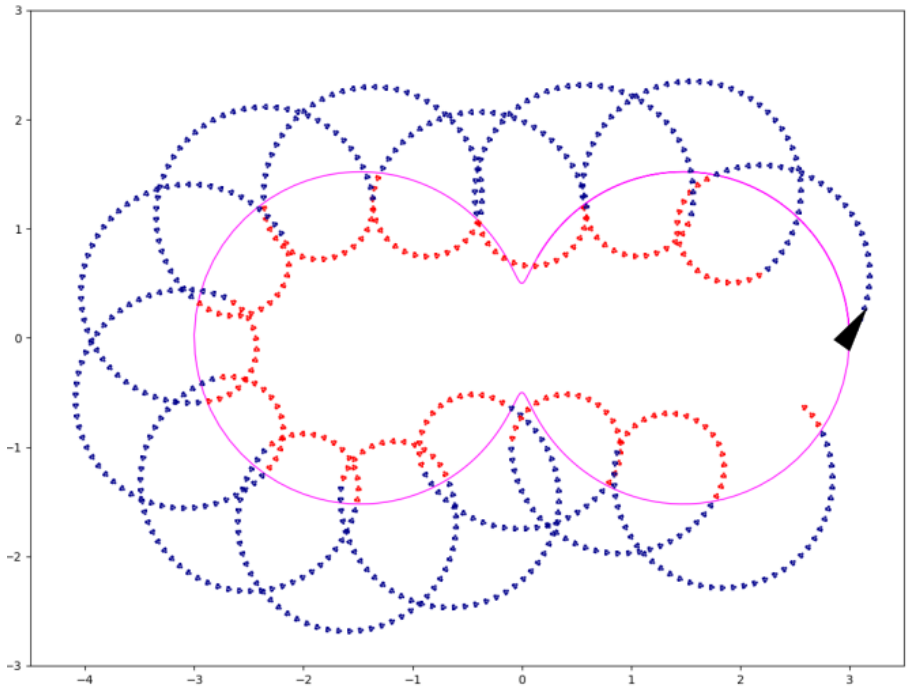


Figure 18: The robot rolls along the Hippopede. The frame box is $[-4.5, 3.5] \times [-3, 3]$.

can be used in an unstructured environment, where few sensors can be used by the robot. Then, we extended our approach to characterize an inner approximation of the attraction domain of the periodic orbit.

The mathematical tools used here were well known [14] for studying attractors of continuous dynamical systems. Our main contribution is the adaptation of these tools, mainly based on the rigorous computation of Poincaré maps, to prove the stability of periodic orbits of hybrid systems. One limitation of the approach is that we have to perform an undefined number of laps before proving the periodic stability. An extension would be the use of ellipsoids instead of boxes. This would allow us to perform only one lap by choosing the right shape for the ellipsoid. Moreover, each time we intersect a surface, the wrapping effect introduced by the intersection would significantly decrease with ellipsoids, since the intersection between one ellipsoid and one plane is still an ellipsoid.

The Python programs associated to all examples can be found here: https://www.ensta-bretagne.fr/jaulin/rolling.html.

References

[1] Arnold, V. *Geometrical Methods in the Theory of Ordinary Differential Equations*. Springer-Verlag, 1988. DOI: 10.1007/978-3-662-11832-0.

[2] Aubin, J.-P. Viability Kernels and Capture Basins of Sets Under Differential Inclusions. *SIAM Journal on Control and Optimization*, 40(3):853–881, 2001. DOI: 10.1137/S036301290036968X.

[3] Aubin, J.P. *Viability Theory*. Birkhäuser Boston, 2009. DOI: 10.1007/978-0-8176-4910-4.

[4] Blanchini, F. and Miani, S. *Set-Theoretic Methods in Control*. Birkhäuser Boston, 2008. DOI: 10.1007/978-0-8176-4606-6.

[5] Bourgois, A. *Safe and Collaborative Autonomous Underwater Docking*. PhD dissertation, Université de Bretagne Occidentale, Brest, France, 2021.

[6] Bourgois, A., Chaabouni, A., Rauh, A., and Jaulin, L. Proving the stability of navigation cycles. In *Proceedings of the 19th International Symposium on Scientific Computing, Computer Arithmetic and Verified Numerical Computation (SCAN 2020)*, pages 16–17, 2021. https://www.inf.u-szeged.hu/scan2020/sites/default/files/scan2020_proceedings.pdf.

[7] Bourgois, A., Chaabouni, A., Rauh, A., and Jaulin, L. *Codes for stability of the rolling experiment*. Robex, Lab-STICC, ENSTA-Bretagne, 2022. https://www.ensta-bretagne.fr/jaulin/rolling.html.

[8] Bourgois, A. and Jaulin, L. Interval centred form for proving stability of non-linear discrete-time systems. *Electronic Proceedings in Theoretical Computer Science*, 331:1–17, 2021. DOI: 10.4204/eptcs.331.1.

[9] Drevelle, V. and Bonnifait, P. Localization confidence domains via set inversion on short-term trajectory. *IEEE Transactions on Robotics*, 29(5):1244–1256, 2013. DOI: 10.1109/tro.2013.2262776.

[10] Dubins, L. E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497, 1957. DOI: 10.2307/2372560.

[11] Girard, A. Computation and stability analysis of limit cycles in piecewise linear hybrid systems. *IFAC Proceedings Volumes*, 36(6):181–186, 2003. DOI: 10.1016/s1474-6670(17)36428-5.

[12] Goldsztejn, A. and Chabert, G. Estimating the robust domain of attraction for non-smooth systems using an interval Lyapunov equation. *Automatica*, 100:371–377, 2019. DOI: 10.1016/j.automatica.2018.03.036.

[13] Hachicho, O. and Tibken, B. Estimating domains of attraction of a class of nonlinear dynamical systems with LMI methods based on the theory of moments. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, Volume 3, pages 3150–3155, 2002. DOI: 10.1109/CDC.2002.1184354.

[14] Kapela, T., Mrozek, M., Wilczak, D., and Zgliczynski, P. CAPD::DynSys: A flexible C++ toolbox for rigorous numerical analysis of dynamical systems. *Commun. Nonlinear Sci. Numer. Simul.*, 101:105578, 2021. DOI: 10.1016/j.cnsns.2020.105578.

[15] Kapela, T., Wilczak, D., and Zgliczynski, P. Recent advances in a rigorous computation of Poincaré maps. *Commun. Nonlinear Sci. Numer. Simul.*, 110:106366, 2022. DOI: 10.1016/j.cnsns.2022.106366.

[16] Kearfott, B. and Kreinovich, V. Applications of interval computations: An introduction. In *Applied Optimization*, pages 1–22. Springer US, 1996. DOI: 10.1007/978-1-4613-3440-8_1.

[17] Le-Mézo, T., Jaulin, L., and Zerr, B. An interval approach to compute invariant sets. *IEEE Transactions on Automatic Control*, 62(8):4236–4242, 2017. DOI: 10.1109/tac.2017.2685241.

[18] Lee, Y. Il and Kouvaritakis, B. Constrained robust model predictive control based on periodic invariance. *Automatica*, 42(12):2175–2181, 2006. DOI: 10.1016/j.automatica.2006.07.004.

[19] Lhommeau, M., Jaulin, L., and Hardouin, L. Capture basin approximation using interval analysis. *International Journal of Adaptive Control and Signal Processing*, 25(3):264–272, 2010. DOI: 10.1002/acs.1195.

[20] Lohner, R. Enclosing the solutions of ordinary initial and boundary value problems. In Kaucher, E., Kulisch, U., and Ullrich, Ch., editors, *Computer Arithmetic: Scientific Computation and Programming Languages*, pages 255–286. BG Teubner, Stuttgart, Germany, 1987.

[21] Lunze, J. and Lamnabhi-Lagarrigue, F. *Handbook of Hybrid Systems Control: Theory, Tools, Applications*. Cambridge University Press, England, 2009. DOI: 10.1017/cbo9780511807930.001.

[22] Moore, R. *Methods and Applications of Interval Analysis*. Society for Industrial and Applied Mathematics, 1979. DOI: 10.1137/1.9781611970906.

[23] Pursche, T., Swiatlak, R., and Tibken, B. Estimation of the domain of attraction for nonlinear autonomous systems using a Bezoutian approach. In *2016 SICE International Symposium on Control Systems (ISCS)*, pages 1–6, 2016. DOI: 10.1109/SICEISCS.2016.7470159.

[24] Ramdani, N. and Nedialkov, N. Computing reachable sets for uncertain nonlinear hybrid systems using interval constraint-propagation techniques. *Nonlinear Analysis: Hybrid Systems*, 5(2):149–162, 2011. DOI: 10.1016/j.nahs.2010.05.010.

[25] Ramdani, N., Travé-Massuyès, L., and Jauberthie, C. Mode discernibility and bounded-error state estimation for nonlinear hybrid systems. *Automatica*, 91:118–125, 2018. DOI: 10.1016/j.automatica.2018.01.022.

[26] Ratschan, S. and She, Z. 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):4377–4394, 2010. DOI: 10.1109/ICCCYB.2006.305705.

[27] Rauh, A. *Sensitivity Methods for Analysis and Design of Dynamic Systems with Applications in Control Engineering*. Shaker–Verlag, 2017. DOI: 10.2370/9783844054989.

[28] Rauh, A., Bourgois, A., and Jaulin, L. Verifying provable stability domains for discrete-time systems using ellipsoidal state enclosures. *Acta Cybernetica*, 2022. DOI: 10.14232/actacyb.293871.

[29] Rauh, A., Bourgois, A., Jaulin, L., and Kersten, J. Ellipsoidal enclosure techniques for a verified simulation of initial value problems for ordinary differential equations. In *2021 International Conference on Control, Automation and Diagnosis (ICCAD)*. IEEE, 2021. DOI: 10.1109/iccad52417.2021.9638755.

[30] Rohou, S. *Reliable robot localization: a constraint programming approach over dynamical systems*. PhD dissertation, Université de Bretagne Occidentale, ENSTA-Bretagne, France, 2017.

[31] Rohou, S., Jaulin, L., Mihaylova, L., Bars, F. Le, and Veres, S. *Reliable Robot Localization*. Wiley, 2019. DOI: 10.1002/9781119680970.

[32] Swiatlak, R., Tibken, B., Paradowski, T., and Dehnert, R. Determination of the optimal quadratic Lyapunov function for nonlinear autonomous systems via interval arithmetic. In *2015 European Control Conference (ECC)*, pages 297–303, 2015. DOI: 10.1109/ECC.2015.7330560.

[33] Tucker, W. The Lorenz attractor exists. *Comptes Rendus de l'Académie des Sciences — Series I — Mathematics*, 328(12):1197–1202, 1999. DOI: 10.1016/s0764-4442(99)80439-x.

[34] Verdu, T., Maury, N., Narvor, P., Seguin, F., Roberts, G., Couvreux, F., Cayez, G., Bronz, M., Hattenberger, G., and Lacroix, S. Experimental flights of adaptive patterns for cloud exploration with UAVs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020, Las Vegas, NV, USA, October 24, 2020 - January 24, 2021*, pages 1429–1435. IEEE, 2020. DOI: 10.1109/IROS45743.2020.9341408.

[35] Wilczak, D. and Zgliczynski, P. Cr-Lohner algorithm. *Schedae Informaticae*, 20:9–46, 2011. DOI: 10.4467/20838476SI.11.001.0287.

[36] Zgliczynski, P. C^1 Lohner algorithm. *Foundations of Computational Mathematics*, 2(4):429–465, 2002. DOI: 10.1007/s102080010025.

Acta Cybernetica **26** (2023) 35–52.

The Inventory Control Problem for a Supply Chain With a Mixed Type of Demand Uncertainty

Elena Chausova[a]

Abstract

This paper is concerned with a dynamic inventory control system described by a network model where the nodes are warehouses and the arcs represent production and distribution activities. We assume that an uncertain demand may take any value in an assigned interval and we allow that the system is disturbed by noise inputs. These assumptions yield a model with a mix of interval and stochastic demand uncertainties. We use the method of model predictive control to derive the control strategy. To deal with interval uncertainty we use the interval analysis tools and act according to the interval analysis theory. The developed results are illustrated using a numerical example.

Keywords: inventory control, supply chain, network model, model predictive control, interval-stochastic uncertainty, interval analysis, multiobjective optimization, quadratic programming

1 Introduction

Nowadays, most supply chains are multi-echelon and have a complex network structure. Such a network consists of suppliers, manufacturing plants, warehouses, customers, and distribution channels that are organized efficiently to get raw materials, convert them to finished products, and distribute the products to customers. The structure of any multi-echelon supply chain depends on the configuration and location of various echelons with respect to each other. It can be described by a directed network in which the nodes represent warehouses and the arcs are controllable and uncontrollable flows in the network. The controllable flows can be controlled by a system manager. They redistribute resources between the network nodes, possibly process them, and plan deliveries from outside. The uncontrollable flows represent a demand in the network nodes that can be made both by other nodes and from the outside. Supply chain managers always seek to find best decisions to provide products or services for customers in the right quantities, at the right places, and at right times.

[a]National Research Tomsk State University, Tomsk, Russia, E-mail: chauev@mail.ru, ORCID: 0000-0003-2379-5224

DOI: 10.14232/actacyb.295044

This paper deals with the inventory control problem in a multi-echelon supply chain network. In the classical inventory control theory uncertainty of demand is regarded as stochastic uncertainty. However, in many real cases, there are not enough historical data to estimate parameters of distributions of random variables that affect the system. This fact gives rise to the need to use other approaches to describing uncertainty. An interesting approach based on unknown-but-bounded inputs is proposed in [2, 3]. The studies are devoted to the inventory control problem under an uncertain demand. Unlike the classical stochastic approach, they model demand uncertainty in an unknown-but-bounded way assuming that an uncertain demand may take any values in an assigned set, and nothing else is known about demand behaviour. This makes sense because in practice the upper and lower bounds for an uncertain demand can be inferred from the decision maker's experience or available historical data much more easily and with much more confidence than empirical probability distribution. At the same time, the efficiency of the control strategy strongly depends on the width of the interval of uncertain demand and this interval should be as narrow as possible. To reduce the width of the interval we can use a mixed form of model uncertainty. This is reasonable, for example, when we have partial information about demands. Indeed, for some products we do not have historical demand data, while for others we do. In addition, we can have quite stable orders within given limits from some consumers and random orders from others. In such cases, an uncertain aggregate demand can be decomposed in two sub-vectors, one of which is unknown-but-bounded (interval), and the other is stochastic. These assumptions lead to a mixed interval-stochastic uncertainty which is used further in this study.

We use the model predictive control [4, 14] (MPC) to derive the optimal control strategy. The MPC approach is widely applied in the practice of control and allows for solving complex control problems for systems with various types of uncertainty. For example, the papers [15, 18] consider supply chain networks under conditions of stochastic uncertainty, and the MPC approach allows the authors to develop a control strategy in order to achieve the system robustness, performance and high levels of service. The paper [1] studies stochastic hybrid systems and shows the effectiveness of suggested techniques for a problem of supply chain management. The paper [8] addresses the problem of the model predictive control for discrete systems with random dependent parameters and its possible application to investment portfolio optimization. The papers [6, 12, 17] examine the MPC problem for systems with a polytopic uncertainty description on state-space matrices under diverse input-output constraints. The problem is solved using the minmax approach to the MPC based on linear matrix inequalities. The paper [7] discusses the case of uncertain linear dynamic systems with interval assigned parameters and multiplicative noises in system matrices. By using the minmax MPC based on linear matrix inequalities, the optimal robust control strategy providing the system with stability in the mean-square sense is obtained. But the lack of constraints does not allow the use of the obtained results for inventory management directly, where, as a rule, there are various capacity constraints. The paper [5] is concerned with the inventory control problem under hard constraints in storage levels and controls. A

linear objective criterion is used to find the optimal control strategy that minimizes the worst-case storage cost under interval demand uncertainty, but the cost of control actions is not taken into account here. The problem is converted into a linear programming problem with constraints to be solved online that gives the optimal control strategy without a shortage and backlogged demand. However, additional stochastic uncertainty is not assumed here.

This paper considers an inventory control system with mixed additive uncertainty in the presence of constraints in the states of the system and control actions. An uncertain demand is estimated by an interval without any distribution information, and the system is assumed to be disturbed by white noise. To deal with the interval uncertainty we use the interval analysis tools and act according to the interval analysis theory [13]. The influence of stochastic uncertainty leads to the minimization of the conditional expected value of the MPC objective under soft constraints in the states of the system. We transform the system control problem with mixed model uncertainty to a deterministic quadratic programming problem for which there are efficient solution methods and commercial software packages (we used the `quadprog` function provided by the software Optimization Toolbox in the MATLAB environment). Solving this problem online, we get a feedback inventory control strategy with a minimum expected level of storage, but a high level of service.

The paper is organized as follows. Section 2 introduces the problem to be solved. Section 3 presents the main results concerning the optimal control under interval-stochastic demand uncertainty. A numerical example showing the results obtained is presented in Section 4 and conclusions are given in Section 5.

2 Problem statement

We consider a dynamic inventory control system with a network structure (supply chain). The evolution of the network is described by the equation:

$$x(k+1) = x(k) + Bu(k) + Cd(k) + Cw(k), \quad k = 0, 1, 2, \ldots \tag{1}$$

Here $x(k) \in \mathbb{R}^n$ is the system state whose components represent storage levels in the network nodes at the time k, the initial state $x(0)$ is assumed to be fixed and given; $u(k) \in \mathbb{R}^m$ is the control representing the controllable flows between the network nodes at the time k; $d(k)$, $w(k) \in \mathbb{R}^l$ are the uncertain demand vectors describing the uncontrollable flows in the network nodes at the time k; the matrices $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{n\times l}$ describe the network structure. As the unit of time k we can take, for example, a day, a week, a month, or a longer period.

Interval uncertainty in the system is represented by the vector $d(k)$. We know that $d(k)$ takes its values within a given interval but the rest is unknown:

$$d(k) \in \boldsymbol{D}, \quad k = 0, 1, 2, \ldots, \tag{2}$$

where $\boldsymbol{D} \in \mathbb{IR}^l$, $\boldsymbol{D} = \left[\underline{D}, \overline{D}\right] \geq 0$.

In the paper we follow the notation of the informal international standard [11]. Intervals and interval objects (vectors, matrices) are denoted in bold, $\underline{x}, \overline{x}$ are the lower and upper bounds of the interval $\boldsymbol{x}$, $\mathbb{IR}^n = \{\boldsymbol{x} = [\underline{x}, \overline{x}], \underline{x} \le \overline{x}, \underline{x}, \overline{x} \in \mathbb{R}^n\}$ is the set of all n-dimensional intervals in the classical interval arithmetic $\mathbb{IR}$, $\mathbb{KR}^n = \{\boldsymbol{x} = [\underline{x}, \overline{x}], \underline{x}, \overline{x} \in \mathbb{R}^n\}$ is the set of all n-dimensional intervals in the Kaucher complete interval arithmetic $\mathbb{KR}$ [10, 16].

The uncertain vector $w(k)$ describes white noise with a zero mean and the covariance matrix $\mathsf{E}\{w(k)w^\top(k)\} = W$. This forms stochastic uncertainty in the system.

Additionally, we assume that both expected storage levels and controls must be non-negative and bounded:

$$\mathsf{E}\big\{x(k+1) \;\big|\; x(k)\big\} \in \boldsymbol{X}, \quad k = 0, 1, 2, \ldots, \tag{3}$$

$$u(k) \in \boldsymbol{U}, \quad k = 0, 1, 2, \ldots, \tag{4}$$

where $\mathsf{E}\{\cdot|\cdot\}$ denotes the conditional expectation; $\boldsymbol{X} \in \mathbb{IR}^n$, $\boldsymbol{X} = \big[0, \overline{X}\big]$; $\boldsymbol{U} \in \mathbb{IR}^q$, $\boldsymbol{U} = \big[0, \overline{U}\big]$. The bounds of the constraints given in (3), (4) define the system's capacities, such as storage limit and order quantity limit. In (3), the lower bound equal to zero means that a shortage of stock is undesirable, but possible. The shortage reduces the service level that is defined as the proportion of demand satisfied. The ideal case is 100% service level. In order to maintain a high service level under the uncertain demand a safety stock is formed. The level of the safety stock for real-life complex, lean, and agile networks can be efficiently determined by the method of the dynamic simulation.

We define the MPC performance index (cost function) as follows:

$$\begin{aligned} J(k+p|k) = \mathsf{E}\Bigg\{ \sum_{i=1}^{p} \Big(& \big(x(k+i|k) - x_0\big)^\top Q \big(x(k+i|k) - x_0\big) \\ & - Q_1\big(x(k+i|k) - x_0\big) + u(k+i-1|k)^\top R u(k+i-1|k)\Big) \;\Big|\; x(k)\Bigg\}, \end{aligned} \tag{5}$$

where $x(k+i|k)$ is the state at the time $k+i$ which is predicted at the time k, $x(k)$ or $x(k|k)$ denotes the state measured at the time k; x_0 is the target level that defines a desired storage level; $u(k+i|k)$ is the predictive control at the time $k+i$ which is computed at the time k; p is the prediction horizon; $Q \in \mathbb{R}^{n\times n}$, $Q_1 \in \mathbb{R}^{1\times n}$ and $R \in \mathbb{R}^{m\times m}$ are the weighting matrices such that Q, R are symmetric positive definite matrices and $Q_1 \ge 0$.

The control goal generally is to keep the state of the system close to the target using little control efforts. But taking into account the fact that we deal with a storage level it is necessary to specify the goal so that the state of the system is close but preferably not below the target level x_0. In cost function (5) the first term $\big(x(k+i|k) - x_0\big)^\top Q\big(x(k+i|k) - x_0\big)$ penalizes the state deviation from the target level, the second linear term $Q_1\big(x(k+i|k) - x_0\big)$ penalizes the state negative deviation from the target level, and the last term $u(k+i-1|k)^\top R u(k+i-1|k)$ penalizes the control efforts.

The problem to be solved is to compute a sequence of the predictive controls $u(k|k), u(k+1|k), \ldots, u(k+p-1|k)$ which minimizes cost index (5):

$$\min_{u(k|k),u(k+1|k),\ldots,u(k+p-1|k)} J(k+p|k),$$

subject to system dynamics (1) and constraints (2), (3), (4).

We reduce the above problem to an interval quadratic programming problem where the uncertain inputs are represented by intervals. Since the input data are interval, the objective value is also interval. We calculate the lower and upper bounds of the objective values of the interval quadratic programming problem analytically using the interval analysis and formulate a two-objective optimization problem. We then transform the problem into a conventional quadratic programming problem with a single objective by using the multi-objective optimization technique [9].

As is standard in the MPC, at the time k we calculate the sequence of predictive controls $u(k|k), u(k+1|k), \ldots, u(k+p-1|k)$, but use only the first of them and obtain the feedback control $u(k) = u(k|k)$ as a function of the state $x(k)$. Then the state $x(k+1)$ is measured, the control horizon is moved by one, and the optimization is repeated at the next time $k+1$. The result is the feedback inventory control strategy $\Phi = \{u(k) = u\,(x(k), k)\,,\ k \geq 0\}$.

3 Main results

The following theorem gives the sequence of predictive controls $\{u(k|k), u(k+1|k), \ldots, u(k+p-1|k)\}$ at the time k.

Theorem. *The vector of predictive controls*

$$\tilde{u}(k) = \left(u(k|k)^\top,\, u(k+1|k)^\top, \ldots,\, u(k+p-1|k)^\top\right)^\top$$

that minimizes performance index (5) *under constraints* (2), (3), (4) *on the trajectories of system* (1) *is defined at each time* k *as a solution to the quadratic programming problem with the criterion*

$$Y(k+p|k) = \tilde{u}(k)^\top H \tilde{u}(k) + 2G(k)\tilde{u}(k) \tag{6}$$

under the constraints

$$\left(B\ 0_{n\times m}\ 0_{n\times m} \ldots 0_{n\times m}\right)\tilde{u}(k) \in \boldsymbol{X} \ominus \boldsymbol{CD} - x(k), \tag{7}$$

$$\tilde{u}(k) \in \tilde{\boldsymbol{U}}. \tag{8}$$

Here $H, G(k)$ *are the block matrices of the type*

$$H = \begin{pmatrix} H_{11} & H_{12} & \ldots & H_{1p} \\ H_{21} & H_{22} & \ldots & H_{2p} \\ \vdots & \ddots & \vdots & \\ H_{p1} & H_{p2} & \ldots & H_{pp} \end{pmatrix}, \quad H_{ij} = \begin{cases} (p-j+1)B^\top QB, & i<j, \\ R + (p-j+1)B^\top QB, & i=j, \\ (p-i+1)B^\top QB, & i>j, \end{cases} \tag{9}$$

$$G(k) = \left(\left(x(k) - x_0\right)^\top Q - \frac{1}{2}Q_1\right) BK + \operatorname{mid} \boldsymbol{DF}, \tag{10}$$

where

$$K = \begin{pmatrix} K_1 & K_2 & \dots & K_p \end{pmatrix}, \quad K_i = (p-i+1)I_m,$$

$$F = \begin{pmatrix} F_{11} & F_{12} & \dots & F_{1p} \\ F_{21} & F_{22} & \dots & F_{2p} \\ \vdots & \ddots & \vdots & \\ F_{p1} & F_{p2} & \dots & F_{pp} \end{pmatrix}, \quad F_{ij} = \begin{cases} (p-j+1)C^\top QB, & i \le j, \\ (p-i+1)C^\top QB, & i > j, \end{cases}$$

$0_{n\times m}$ *is the zero matrix of the dimension* $n \times m$, I_m *is the unit matrix of the dimension* m, $\tilde{\boldsymbol{U}} = \left(\boldsymbol{U}^\top \boldsymbol{U}^\top \dots \boldsymbol{U}^\top\right)^\top$, $\tilde{\boldsymbol{D}} = \left(\boldsymbol{D}^\top \boldsymbol{D}^\top \dots \boldsymbol{D}^\top\right)^\top$, $\boldsymbol{CD}$ *is the result of multiplying the real matrix* C *by the interval vector* $\boldsymbol{D}$, $\boldsymbol{DF}$ *is the result of multiplying the interval vector* $\tilde{\boldsymbol{D}}^\top$ *by the real matrix* F, $\operatorname{mid}\boldsymbol{x}$ *is the midpoint of the interval* $\boldsymbol{x}$, $\boldsymbol{x} \ominus \boldsymbol{y} = \left[\underline{x} - \underline{y}, \overline{x} - \overline{y}\right]$ *is the internal subtraction in* $\mathbb{KR}$.

Proof. Let us consider performance index (5). By using the fact that the summand

$$\begin{aligned} &(x(k+i|k) - x_0)^\top Q\,(x(k+i|k) - x_0) - Q_1\,(x(k+i|k) - x_0) \\ &+ u(k+i-1|k)^\top Ru(k+i-1|k) = x(k+i|k)^\top Qx(k+i|k) - \left(2x_0^\top Q + Q_1\right) x(k+i|k) \\ &\qquad + \left(x_0^\top Q + Q_1\right) x_0 + u(k+i-1|k)^\top Ru(k+i-1|k), \end{aligned}$$

(5) turns into

$$\begin{aligned} J(k+p|k) = \mathsf{E}\Big\{ \sum_{i=1}^{p} \Big(& x(k+i|k)^\top Qx(k+i|k) - \left(2x_0^\top Q + Q_1\right) x(k+i|k) \\ & + u(k+i-1|k)^\top Ru(k+i-1|k) \Big) \;\Big|\; x(k) \Big\} + p\left(x_0^\top Q + Q_1\right) x_0. \end{aligned}$$

To deal with the conditional expectation, we rewrite the index as:

$$\begin{aligned} J(k+p|k) = \mathsf{E}\Big\{ & x(k+1|k)^\top Qx(k+1|k) - \left(2x_0^\top Q + Q_1\right) x(k+1|k) \\ & + u(k|k)^\top Ru(k|k) + \mathsf{E}\Big\{ x(k+2|k)^\top Qx(k+2|k) \\ & - \left(2x_0^\top Q + Q_1\right) x(k+2|k) + u(k+1|k)^\top Ru(k+1|k) + \dots \\ & + \mathsf{E}\big\{ x(k+p|k)^\top Qx(k+p|k) - \left(2x_0^\top Q + Q_1\right) x(k+p|k) \\ & + u(k+p-1|k)^\top Ru(k+p-1|k) \;\big|\; x(k+p-1) \big\} \dots \\ & \Big|\; x(k+1) \Big\} \;\Big|\; x(k) \Big\} + p\left(x_0^\top Q + Q_1\right) x_0. \end{aligned}$$

We introduce the notation

$$\begin{aligned}
J_{k+i} = \mathsf{E}\Big\{ & x(k+i+1|k)^\top Q x(k+i+1|k) - \left(2x_0^\top Q + Q_1\right) x(k+i+1|k) \\
& + u(k+i|k)^\top R u(k+i|k) + \mathsf{E}\Big\{ x(k+i+2|k)^\top Q x(k+i+2|k) \\
& - \left(2x_0^\top Q + Q_1\right) x(k+i+2|k) + u(k+i+1|k)^\top R u(k+i+1|k) + \dots \\
& + \mathsf{E}\big\{ x(k+p|k)^\top Q x(k+p|k) - \left(2x_0^\top Q + Q_1\right) x(k+p|k) \\
& + u(k+p-1|k)^\top R u(k+p-1|k) \;\big|\; x(k+p-1) \big\} \dots \\
& \Big| \; x(k+i+1) \Big\} \;\Big|\; x(k+i) \Big\}.
\end{aligned}$$

Now it is clear that

$$\begin{aligned}
J_{k+i} = \mathsf{E}\big\{ & x(k+i+1|k)^\top Q x(k+i+1|k) - \left(2x_0^\top Q + Q_1\right) x(k+i+1|k) \\
& + u(k+i|k)^\top R u(k+i|k) + J_{k+i+1} \;\big|\; x(k+i) \big\}, \quad i = 0, \dots, p-1,
\end{aligned} \tag{11}$$

with $J_{k+p} = 0$ and

$$J(k+p|k) = J_k + p\left(x_0^\top Q + Q_1\right) x_0. \tag{12}$$

Using the method of mathematical induction we prove that the following relationship is valid:

$$\begin{aligned}
J_{k+p-t} = {} & t x(k+p-t|k)^\top Q x(k+p-t|k) \\
& + \Big(2\big(x(k+p-t|k) - x_0\big)^\top Q - Q_1\Big) \\
& \times \left(\sum_{i=1}^{t} iBu(k+p-i|k) + \sum_{i=1}^{t} iCd(k+p-i) \right) \\
& + \sum_{i=1}^{t} u(k+p-i|k)^\top \left(2 \sum_{j=1}^{i-1} jB^\top Q B u(k+p-j|k) \right. \\
& \left. + (iB^\top Q B + R) u(k+p-i|k) \right) + 2 \sum_{i=1}^{t} \left(\sum_{j=1}^{i} jd(k+p-j)^\top \right. \\
& \left. + \sum_{j=i+1}^{t} id(k+p-j)^\top \right) C^\top Q B u(k+p-i|k) \\
& + \sum_{i=1}^{t} d(k+p-i)^\top C^\top Q C \left(2 \sum_{j=1}^{i-1} jd(k+p-j) + id(k+p-i) \right) \\
& + \operatorname{tr}\left\{ \frac{t(t+1)}{2} C^\top Q C W \right\} - t(2x_0^\top Q + Q_1) x(k+p-t|k), \quad t = 1, \dots, p,
\end{aligned} \tag{13}$$

where $\operatorname{tr}\{\cdot\}$ is the trace of a matrix.

At first, let us consider the case for $p = 1$. From (11) for $i = p - 1$ we have

$$J_{k+p-1} = \mathsf{E}\big\{x(k+p|k)^\top Qx(k+p|k) - \big(2x_0^\top Q + Q_1\big)\, x(k+p|k) + u(k+p-1|k)^\top Ru(k+p-1|k) \;\big|\; x(k+p-1)\big\}. \tag{14}$$

Substituting $x(k+p|k)$ by its expression in terms of $x(k+p-1|k)$ from (1) in (14) and taking the conventional mathematical expectation, we get

$$\begin{aligned} J_{k+p-1} &= x(k+p-1|k)^\top Qx(k+p-1|k) + \Big(2\big(x(k+p-1|k) - x_0\big)^\top Q - Q_1\Big) \\ &\times \big(Bu(k+p-1|k) + Cd(k+p-1)\big) \\ &+ u(k+p-1|k)^\top (B^\top QB + R)u(k+p-1|k) \\ &+ 2d(k+p-1)^\top C^\top QBu(k+p-1|k) + d(k+p-1)^\top C^\top QCd(k+p-1) \\ &+ \operatorname{tr}\{C^\top QCW\} - \big(2x_0^\top Q + Q_1\big)\, x(k+p-1|k), \end{aligned}$$

and this coincides with (13) if $t = 1$.
Now let us suppose that relationship (13) is valid for some t, and show that (13) is valid for $t + 1$. Indeed, from recursive expression (11) we obtain

$$J_{k+p-t-1} = \mathsf{E}\big\{x(k+p-t|k)^\top Qx(k+p-t|k) - \big(2x_0^\top Q + Q_1\big)\, x(k+p-t|k) + u(k+p-t-1|k)^\top Ru(k+p-t-1|k) + J_{k+p-t} \;\big|\; x(k+p-t)\big\}. \tag{15}$$

Now we will substitute $x(k+p-t|k)$ by its expression in terms of $x(k+p-t-1|k)$ from (1) in (15) and J_{k+p-t} by its expression from (13). Expanding the conventional expectation and transforming the expression, we obtain that

$$\begin{aligned} J_{k+p-t-1} &= (t+1)x(k+p-t-1|k)^\top Qx(k+p-t-1|k) \\ &+ \Big(2\big(x(k+p-t-1|k) - x_0\big)^\top Q - Q_1\Big) \\ &\times \left(\sum_{i=1}^{t+1} iBu(k+p-i|k) + \sum_{i=1}^{t+1} iCd(k+p-i)\right) \\ &+ \sum_{i=1}^{t+1} u(k+p-i|k)^\top \left(2\sum_{j=1}^{i-1} jB^\top QBu(k+p-j|k)\right. \\ &\left. + (iB^\top QB + R)u(k+p-i|k)\right) + 2\sum_{i=1}^{t+1}\left(\sum_{j=1}^{i} jd(k+p-j)^\top\right. \\ &\left. + \sum_{j=i+1}^{t+1} id(k+p-j)^\top\right) C^\top QBu(k+p-i|k) \\ &+ \sum_{i=1}^{t} d(k+p-i)^\top C^\top QC\left(2\sum_{j=1}^{i-1} jd(k+p-j) + id(k+p-i)\right) \\ &+ \operatorname{tr}\left\{\frac{(t+1)(t+2)}{2} C^\top QCW\right\} - (t+1)(2x_0^\top Q + Q_1)x(k+p-t-1|k). \end{aligned} \tag{16}$$

Formula (16) coincides with (13) if t is replaced by $t+1$, and hence, according to the mathematical induction rule, formula (13) is valid for all $t = 1, \ldots, p$.
Using the fact that (13) gives an expression for J_k with $t = p$, we get from (12):

$$\begin{aligned} J(k+p|k) &= px(k|k)^\top Qx(k|k) + \Big(2\big(x(k|k) - x_0\big)^\top Q - Q_1\Big) \\ &\times \left(\sum_{i=1}^{p} iBu(k+p-i|k) + \sum_{i=1}^{p} iCd(k+p-i)\right) \\ &+ \sum_{i=1}^{p} u(k+p-i|k)^\top \left(2\sum_{j=1}^{i-1} jB^\top QBu(k+p-j|k)\right. \\ &\left. + \big(iB^\top QB + R\big)u(k+p-i|k)\right) + 2\sum_{i=1}^{p}\Big(\sum_{j=1}^{i} jd(k+p-j)^\top \\ &+ \sum_{j=i+1}^{p} id(k+p-j)^\top\Big)C^\top QBu(k+p-i|k) \\ &+ \sum_{i=1}^{p} d(k+p-i)^\top C^\top QC\left(2\sum_{j=1}^{i-1} jd(k+p-j) + id(k+p-i)\right) \\ &+ \operatorname{tr}\left\{\frac{p(p+1)}{2}C^\top QCW\right\} - p\left(2x_0^\top Q + Q_1\right)x(k|k) + p\left(x_0^\top Q + Q_1\right)x_0. \end{aligned}$$

Eliminating all the terms that do not depend on the controls u and do not influence the optimum, we obtain

$$\begin{aligned} \mathcal{J}(k+p|k) &= \Big(2\big(x(k|k) - x_0\big)^\top Q - Q_1\Big)\sum_{i=1}^{p} iBu(k+p-i|k) \\ &+ \sum_{i=1}^{p} u(k+p-i|k)^\top \left(2\sum_{j=1}^{i-1} jB^\top QBu(k+p-j|k)\right. \\ &\left. + \big(iB^\top QB + R\big)u(k+p-i|k)\right) + 2\sum_{i=1}^{p}\Big(\sum_{j=1}^{i} jd(k+p-j)^\top \\ &+ \sum_{j=i+1}^{p} id(k+p-j)^\top\Big)C^\top QBu(k+p-i|k). \end{aligned} \tag{17}$$

Expression (17) can be rewritten in a matrix form as:

$$\mathcal{J}(k+p|k) = \tilde{u}(k)^\top H\tilde{u}(k) + 2\mathcal{G}(k)\tilde{u}(k), \tag{18}$$

where $\tilde{u}(k) = \big(u(k|k)^\top,\ u(k+1|k)^\top, \ldots,\ u(k+p-1|k)^\top\big)^\top$, H is given by (9),

$$\mathcal{G}(k) = \left(\big(x(k) - x_0\big)^\top Q - \frac{1}{2}Q_1\right)BK + \tilde{d}(k)^\top F,$$

and

$$\tilde{d}(k) = \big(d(k)^\top, d(k+1)^\top, \dots, d(k+p-1)^\top\big)^\top, \ \tilde{d}(k) \in \tilde{\boldsymbol{D}}. \tag{19}$$

Now we will consider constraints (3), (4). It is clear that (4) leads to constraint (8). Using the expression in terms of $x(k)$ from (1) instead of $x(k+1)$ in (3) we get

$$\begin{aligned} \mathsf{E}\big\{x(k+1) \bigm| x(k)\big\} &= \mathsf{E}\big\{x(k) + Bu(k) + Cd(k) + Cw(k) \bigm| x(k)\big\} \\ &= x(k) + Bu(k) + Cd(k) \in \boldsymbol{X}. \end{aligned} \tag{20}$$

Under constraint (2), condition (20) turns into the next inclusion:

$$x(k) + Bu(k) + \boldsymbol{CD} \in \boldsymbol{X},$$

that is valid if and only if

$$x(k) + Bu(k) \in \boldsymbol{X} \ominus \boldsymbol{CD}.$$

Then for the observed state of the system $x(k)$ at each sampling time k the control $u(k)$ must satisfy

$$Bu(k) \in \boldsymbol{X} \ominus \boldsymbol{CD} - x(k),$$

which is consistent with (7).

We come to the problem of minimizing quadratic function (18) with interval data (19) subject to constraints (7), (8). To handle the interval data in (18) we convert the problem of interval quadratic programming into the following two-objective optimization problem:

$$\begin{aligned} &\min_{\tilde{u}(k)} \underline{\mathcal{J}}(k+p|k) = \tilde{u}(k)^\top H \tilde{u}(k) + 2\underline{\mathcal{G}}(k)\tilde{u}(k), \\ &\min_{\tilde{u}(k)} \overline{\mathcal{J}}(k+p|k) = \tilde{u}(k)^\top H \tilde{u}(k) + 2\overline{\mathcal{G}}(k)\tilde{u}(k), \\ &\qquad\qquad \text{subject to (7), (8),} \end{aligned} \tag{21}$$

where the first objective function is the lower bound of interval quadratic function (18) over the interval $\tilde{\boldsymbol{D}}$, and the second is its upper bound. In (21)

$$\begin{aligned} \underline{\mathcal{G}}(k) &= \Big(\big(x(k) - x_0\big)^\top Q - \frac{1}{2} Q_1\Big) BK + \underline{DF}, \\ \overline{\mathcal{G}}(k) &= \Big(\big(x(k) - x_0\big)^\top Q - \frac{1}{2} Q_1\Big) BK + \overline{DF}, \end{aligned}$$

and $\underline{DF}, \overline{DF}$ are the lower and upper bounds of the possible values of $\tilde{d}(k)^\top F$ over the interval $\tilde{\boldsymbol{D}}$.

According to the multi-objective optimization technique [9], problem (21) can be transformed into a quadratic programming problem with a single objective. Based on the scalarization method (the weighting objectives method), we obtain

an equivalent compromise single objective optimisation problem where the objective is chosen as a weighted sum of the original criteria:

$$\min_{\tilde{u}(k)} Y(k+p|k) = \lambda_1 \underline{\mathcal{J}}(k+p|k) + \lambda_2 \overline{\mathcal{J}}(k+p|k)$$
$$\text{subject to (7), (8),}$$

where $\lambda_1, \lambda_2 \geq 0$ are the weighting coefficients that represent the relative importance of each criterion, $\lambda_1 + \lambda_2 = 1$. At various weights, we can express varies preferences to estimate the performance objective. For example, $\lambda_1 = 1$ means the optimistic estimate, $\lambda_2 = 1$ states the pessimistic estimate, $\lambda_1 = \lambda_2 = 0.5$ indicates the neutral estimate. It can be tuned manually until the controller reflects the desired behaviour. From our experience, if the demand values are more or less evenly distributed within its intervals, the equal weights give quite good results. Assuming equal weights in objective functions (21) we obtain the following result:

$$\begin{aligned} Y(k+p|k) &= \left(\underline{\mathcal{J}}(k+p|k) + \overline{\mathcal{J}}(k+p|k)\right)/2 \\ &= \tilde{u}(k)^\top H \tilde{u}(k) + 2\frac{1}{2}\left(\underline{\mathcal{G}}(k) + \overline{\mathcal{G}}(k)\right)\tilde{u}(k) = \tilde{u}(k)^\top H \tilde{u}(k) \\ &+ 2\Big(\big((x(k) - x_0)^\top Q - \frac{1}{2}Q_1\big)BK + \frac{1}{2}\left(\underline{DF} + \overline{DF}\right)\Big)\tilde{u}(k), \end{aligned}$$

that is consistent with (6) and (10). □

At this point, it is worth noting that, due to the interval uncertainty in the system, we can only steer the state to a tube sufficiently close to the target level x_0, and keep the state trajectory on average within the target tube. The target tube is a sequence of the sets that at each time contain all the states whose future trajectory can be kept inside the constraints, for all admissible disturbances [2]. It is clear, the width of this tube depends on the width of the initial uncertainty intervals. Indeed, the problem of keeping the state $x(k)$, on average, in some tube $\boldsymbol{X}(a,b) = [a,b]$ has a solution if and only if, for all $x(k) \in \boldsymbol{X}(a,b)$, there is a control $u(t) \in \boldsymbol{U}$ so that

$$\mathsf{E}\{x(k+1) \mid x(k)\} = x(k) + Bu(k) + Cd(k) \in \boldsymbol{X}(a,b)$$

is valid for all $d(k) \in \boldsymbol{D}$. That takes place if and only if

$$x(k) + Bu(k) + \boldsymbol{CD} \in \boldsymbol{X}(a,b),$$

and then

$$x(k) + Bu(k) \in \boldsymbol{X}(a,b) \ominus \boldsymbol{CD}.$$

It makes sense if and only if $\boldsymbol{X}(a,b) \ominus \boldsymbol{CD} \in \mathbb{IR}$, that is $a - \underline{CD} \leq b - \overline{CD}$. We can argue that $\overline{CD} - \underline{CD} \leq b - a$, and $\operatorname{wid} \boldsymbol{CD} \leq \operatorname{wid} \boldsymbol{X}(a,b)$. Therefore, the minimum width of the tube, within which on average the state $x(k)$ can be kept for all possible values of the demand, is given by

$$\operatorname{wid} \boldsymbol{CD} = \overline{CD} - \underline{CD}.$$

Evidently, under an excessive storage level any system must pay a high storage cost. But if the storage level is too low, the system will have a low service level due to the shortage, resulting in lost profits and loss of customer loyalty. To find the trade-off, we need to maintain the minimum level of storage without violating state constraints for all possible realizations of model uncertainty. This is why we suggest setting the target level at zero during the first simulation and waiting for the tube $\boldsymbol{X}(0, \operatorname{wid} \boldsymbol{CD})$ to be received. In this case, the control is obtained by pointing to the order-up-to-level in the sense that

$$x(k) + Bu(k) = -\underline{CD} \tag{22}$$

because of $\boldsymbol{X}(0, \operatorname{wid} \boldsymbol{CD}) \ominus \boldsymbol{CD} = -\underline{CD}$. Thus, the developed feedback control turns out to be a periodic review, order-up-to-level (R, S) strategy, where the review interval R is the unit of time, and the order-up-to-level S is equal to $-\underline{CD}$. If the levels of service in the network nodes are high enough, there is no need to raise the target level x_0. Otherwise, we can gradually increase the target level and form a safety stock until the required levels of service are received.

4 Numerical Problem

Now we will apply the results obtained in Section 3 to an example. Let us consider the fictional production-distribution system represented by Figure 1.

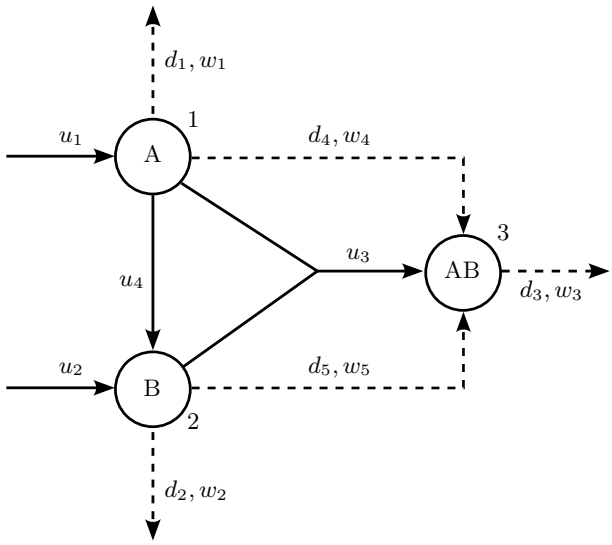


Figure 1: The network structure of a production-distribution system with three nodes and controllable (solid) and uncontrollable (dashed) flows between them

The system has three interdependent production-distribution centres, represented by three nodes. Nodes 1 and 2 make products A and B, these products are

used later for making product AB in node 3. The controllable flows $u_1,\ u_2$ describe the production levels of A in node 1 and B in node 2, respectively, per unit of time, u_3 describes a production line in node 3 which takes some amount of products A and B to produce the same amount AB in node 3. The arc u_4 models an additional flexible capacity present in the system which can be split in any proportion between two production lines A and B. If the arc u_4 works at full force, the flexible capacity is fully used to produce B, while if it works at zero force, the flexible capacity is fully used to produce A. The uncontrollable flows represent the demand in the network nodes that can arise from outside and other nodes. The arcs $d_1,\ d_2,\ d_3$ represent demands for products A, B and AB. And there are the redistribution arcs $d_4,\ d_5$ which represent demands that may unpredictably require A or AB, and B or AB, respectively.
The structural matrices B and C for the system have the form:

$$B = \begin{pmatrix} 1 & 0 & -1 & -1 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 1 \end{pmatrix}.$$

The constraints in the states and controls are given as follows

$$\boldsymbol{X} = \begin{pmatrix} [0,130] & [0,120] & [0,150] \end{pmatrix}^\top,$$

$$\boldsymbol{U} = \begin{pmatrix} [0,170] & [0,50] & [0,100] & [0,70] \end{pmatrix}^\top.$$

The demand $d(k)$ takes values within the interval vector

$$\boldsymbol{D} = \begin{pmatrix} [5,25] & [20,30] & [60,80] & [0,20] & [0,10] \end{pmatrix}^\top.$$

This example is an adapted version of the example from [2]. The system contains the white noise $w(k)$ with a zero mean and the covariance matrix

$$W = \operatorname{diag}\left(\sigma_1^2, \sigma_2^2, \ldots, \sigma_l^2\right),\ \sigma_i^2 = 0.25 \operatorname{wid} \boldsymbol{D}_i.$$

We assume that the demand cannot be backlogged and that demands during stock-outs are completely lost. The initial storage level is $x(0) = (130\ \ 120\ \ 150)^\top$ and the target storage level is $x_0 = (0\ 0\ 0)^\top$. The weighting matrices are chosen as $Q = I_n, Q_1 = (1\ 1\ 1)^\top, R = I_m$, the prediction horizon is $p = 6$, the problem is solved for 100 time steps. We carried out modelling and simulation in MATLAB. The simulation results are presented in Figures 2, 3, 4, 5.

Figure 2 shows the time behaviour of demands in the network. Normally, they fluctuate inside the given intervals, but there are some peaks lying outside their lower and upper bounds. This is the influence of random disturbances that can cause the demand to leave the predicted interval. We take them into account only in the expected way, and this is reflected in customer service levels. But in our case, decrease in the service levels is insignificant. As the simulation showed, we received high levels of service in the network nodes. They are maintained at the

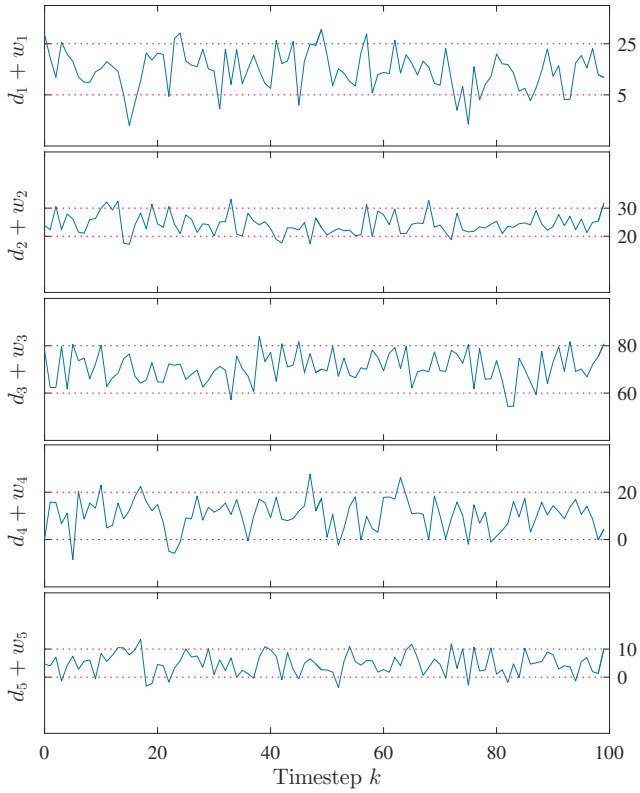


Figure 2: The dynamics of $d_i + w_i$ (solid blue) and the lower and upper bounds of the demand intervals $\left[\underline{D_i}, \overline{D_i}\right]$ (dashed red), $i = 1, \ldots, 5$

level of 98.72% in node 1, 99.98% in node 2, and 99.67% in node 3. In this case, there is no need to increase the target level x_0 to form a safety stock.

Figure 3 demonstrates the controls in the network. The average time required to compute the control actions within a time step using the `quadprog` function was about 0.005 seconds. It is worth noting that the arc u_2 works at full force. The flexible capacity is divided between the production lines A and B ($u_4 > 0$). This means that the constraint in u_2 is limiting.

Figure 4 presents the inventory dynamics in the network nodes under the optimal control strategy. In all the nodes, a decreasing trend of the storage levels can be observed. In our case, $\boldsymbol{CD} = ([-45, -5]\,[-40, -20]\,[-80, -30])^\top$ and $\operatorname{wid}\boldsymbol{CD} = (40\ \ 20\ \ 50)^\top$. Figure 4 shows that starting from some timestep, the state trajectory on average lies within the minimal tube $\boldsymbol{X}(0, \operatorname{wid}\boldsymbol{CD})$.

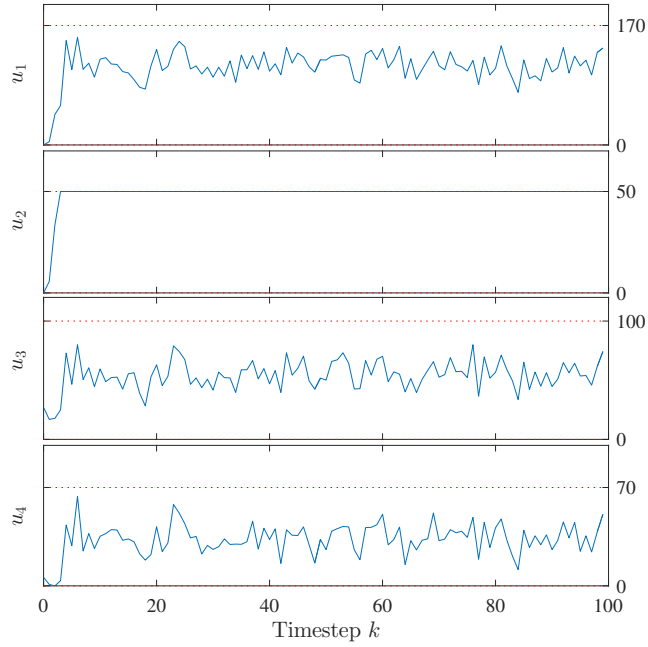


Figure 3: The trajectories of the controls u_1, u_2, u_3, u_4 (solid blue) and its constraints (dashed red)

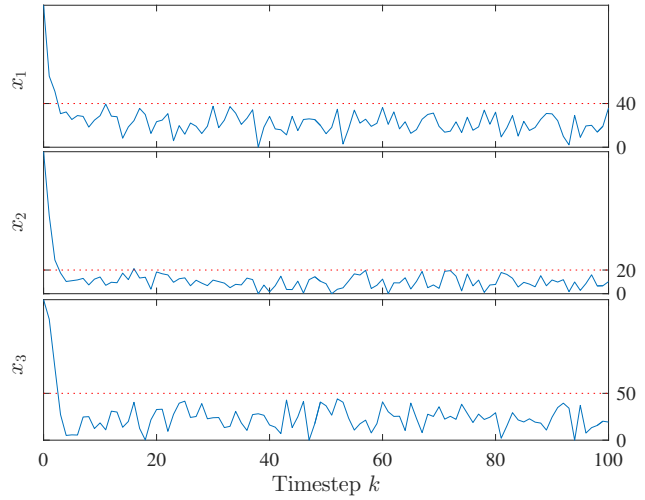


Figure 4: The trajectories of the states x_i (solid blue) and the levels wid $\boldsymbol{CD}_i$ (dashed red), $i = 1, 2, 3$

Figure 5 shows the order-up-to-levels which starting from some point in time are constant and equal to $-\underline{CD} = (45 \;\; 40 \;\; 80)^\top$. This fact is consistent with (22).

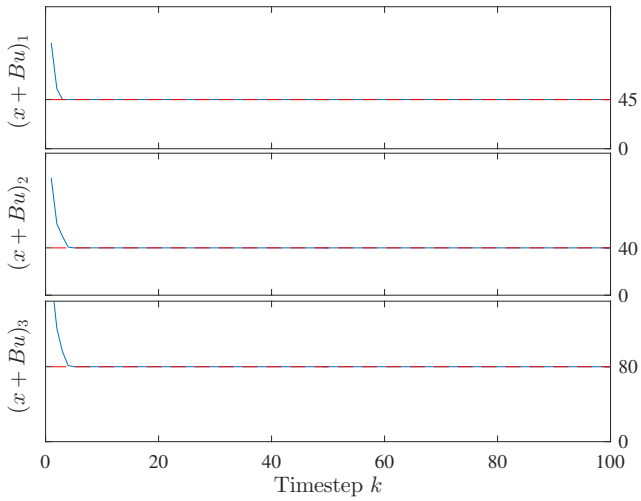


Figure 5: The order-up-to-levels (solid blue) and $-\underline{CD}$ (dashed red)

5 Conclusions and further research

In this study, we considered a supply chain network under interval and stochastic uncertainties. The mixed type of uncertainty is preferred in many cases since it is close to real life. We used the integrated approach to inventory control, with all the network nodes optimized simultaneously. We applied the MPC approach and reduced the problem to a constrained quadratic programming problem which can be solved using efficient techniques. As a result, we developed a feedback inventory control strategy with a high level of service.

However, there are still a number of issues that need to be addressed, such as the case of nonstationary demand, multiplicative noise, storage loss, and the conditions for the existence of controls to fulfill any values of possible demands under interval-stochastic uncertainty. These are the points of possible future research.

References

[1] Bemporad, A. and Di Cairano, S. Model-predictive control of discrete hybrid stochastic automata. *IEEE Transactions on Automatic Control*, 56(6):1307–1321, 2011. DOI: 10.1109/TAC.2010.2084810.

[2] Blanchini, F., Rinaldi, F., and Ukovich, W. Least inventory control of multi-storage systems with non-stochastic unknown demand. *IEEE Transaction on Robotics and Automation*, 13(5):633–645, 1997. DOI: 10.1109/70.631225.

[3] Blanchini, F., Rinaldi, F., and Ukovich, W. A network design problem for a distribution system with uncertain demands. *SIAM Journal on Optimization*, 7(2):560–578, 1997. DOI: 10.1137/S1052623494266262.

[4] Camacho, E.F. and Bordons, C. *Introduction to Model Predictive Control. In: Model Predictive control. Advanced Textbooks in Control and Signal Processing.* Springer, London, 2007. DOI: 10.1007/978-0-85729-398-5_1.

[5] Chausova, E.V. Dynamic network inventory control model with interval non-stationary demand uncertainty. *Numerical Algorithms*, 37(1-4 SPEC. ISS.):71–84, 2004. DOI: 10.1023/B:NUMA.0000049457.89377.12.

[6] Cuzzola, A.F., Geromel, J.C., and Morari, M. An improved approach for constrained robust model predictive control. *Automatica*, 38(7):1183–1189, 2002. DOI: 10.1016/S0005-1098(02)00012-2.

[7] Dombrovskii, V.V. and Chausova, E.V. Model predictive control for linear systems with interval and stochastic uncertainties. *Reliable Computing*, 19(4):351–360, 2014. https://www.reliable-computing.org/reliable-computing-19-pp-351-360.pdf.

[8] Dombrovskii, V.V., Dombrovskii, D.V., and Lashenko, E.A. Model predictive control of systems with random dependent parameters under constraints and its application to the investment portfolio optimization. *Autom Remote Control*, 67:1927–1939, 2006. DOI: 10.1134/S000511790612006X.

[9] Gunantara, N. A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1), 2018. DOI: 10.1080/23311916.2018.1502242.

[10] Kaucher, E. *Interval analysis in the extended interval space* $\mathbb{IR}$. In *Fundamentals of Numerical Computation (Computer-Oriented Numerical Analysis). Computing Suppl.*, Volume 2, pages 33–49. Springer, Vienna, 1980. DOI: 10.1007/978-3-7091-8577-3_3.

[11] Kearfott, R.B., Nakao, M.T., Neumaier, A., Rump, S.M., Shary, S.P., and van Hentenryck, P. Standardized notation in interval analysis. *Computational Technologies*, 15(1):7–13, 2010. https://www.researchgate.net/publication/268495097_Standardized_notation_in_interval_analysis.

[12] Kothare, M.V., Balakrishnan, V., and Morari, M. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996. DOI: 10.1016/0005-1098(96)00063-5.

[13] Moore, Ramon E., Kearfott, R. Baker, and Cloud, Michael J. *Introduction to Interval Analysis.* SIAM, Philadelphia, PA, 2009. DOI: 10.1137/1.9780898717716.

[14] Schwenzer, M., Ay, M., Bergs, T., and Abel, D. Review on model predictive control: An engineering perspective. *Int J Adv Manuf Technol*, 117:1327–1349, 2021. DOI: 10.1007/s00170-021-07682-3.

[15] Seferlis, P. and Giannelos, N. A two-layered optimisation-based control strategy for multi-echelon supply chain networks. *Computers and Chemical Engineering*, 28(5):799–809, 2004. DOI: 10.1016/j.compchemeng.2004.02.022.

[16] Shary, S.P. Numerical computation of formal solutions to interval linear systems of equations. DOI: 10.48550/arXiv.1903.10272, Submitted in arxiv.org/abs/1903.10272v1 [math.NA], 15 Mar 2019.

[17] Wan, Z. and Kothare, M.V. Efficient robust constrained model predictive control with a time varying terminal constraint set. *Systems and Control Letters*, 48:375–383, 2003. DOI: 10.1016/S0167-6911(02)00291-8.

[18] Wang, W. and Rivera, D.E. Model predictive control for tactical decision-making in semiconductor manufacturing supply chain management. *IEEE Transactions on Control Systems Technology*, 16(5):841–855, 2008. DOI: 10.1109/TCST.2007.916327.

Acta Cybernetica **26** (2023) 53–69.

Inverses of Rational Functions*

Tamas Dozsa[a]

Abstract

We consider the numerical construction of inverses for a class of rational functions. We propose two inverse algorithms, which can be used to simultaneously identify every zero of a rational function or polynomial. In the first case, we propose a generalization of an inverse algorithm based on our previous work and specify a class of rational functions, for which this generalized algorithm is applicable. In the second case, we provide a method to construct Blaschke-products, whose roots match the roots of a polynomial or a rational function. We also consider different iterative methods to numerically calculate the inverse points and discuss their properties.

Keywords: rational functions, Blaschke-products, fixed point iterations, winding numbers

1 Introduction

Rational functions play a crucial role in many theoretical and engineering applications. Rational orthogonal systems, such as the Malmquist-Takenaka system were proven to be well suited for several biological signal processing tasks [8, 13]. The transfer functions of linear systems are also rational, making the study of rational functions essential in system identification [12, 14]. Special types of rational functions, such as Blaschke-products also form the basis of many theoretical applications such as the Riesz-Nevanlinna factorization of Hardy-spaces [13], hyperbolic wavelet construction [12] and the construction of bi-orthogonal systems [6].

Our objective in this paper is to describe and numerically produce all solutions of the implicit equation

$$f(\phi) = \Gamma \subset \mathbb{C}, \tag{1}$$

*Supported by the ÚNKP-21-3 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

[a]Department of Numerical Analysis, Eötvös Loránd University, Budapest, Hungary and Systems and Control Laboratory, SZTAKI, Budapest, Hungary, E-mail: dotuaai@inf.elte.hu, dozsatamas@sztaki.hu, ORCID: 0000-0003-0919-4385

DOI: 10.14232/actacyb.294975

where $f \in \mathcal{R}$ belongs to a class of rational functions and Γ is a simple smooth curve. The proposed methods aim to generalize our previous results for Blaschke-products described in [4, 5]. In [5], we provided sufficient conditions on Γ for the distinct, continuous solutions $\phi_k,\ k = 1, \ldots, n$ to uniquely exist if f is an n-factor Blaschke-product. Furthermore, we proposed an inverse algorithm, which can be used to find all solutions of (1).

The rest of this paper is organized as follows. In Section 2 we provide sufficient conditions on Γ for the existence of distinct continuous solutions to (1), if f is rational. In Section 3, we specify the class of considered rational functions $\mathcal{R}$, and propose a generalization of the inverse algorithm in [5] to produce the inverses of any $f \in \mathcal{R}$. Alternatively, one can construct a Blaschke-product, whose zeros match the zeros of the function in question and apply the algorithm proposed in [5] as-is. This approach and its properties are discussed in Section 3.5. In Section 4 we consider different numerical iterative methods and highlight their advantages for use with the proposed, generalized inverse algorithm. Finally, we summarize our results in Section 5.

2 Inverses of Analytic Functions along a Curve

In this section we will discuss the inverses of functions along a curve. Let f be an analytic function on the region $\Omega \subset \mathbb{C}$ and denote by $\Omega' := f(\Omega)$ its range. Furthermore, let $K := \{z \in \Omega : f'(z) = 0\}$ be the set of critical points and $K' = f(K)$ their image with respect to f. We note that, if f is a polynomial every point in K falls into the convex hull of the roots of f [11], while if f happens to be a Blaschke-product, all of its critical points fall into the hyperbolic convex hull of its zeros [11].

The analytic function f can be locally inverted in any $z_0 \in \Omega \setminus K$ [7, 11]. In other words, for any $W_0 \subset \Omega'$ neighborhood of the point $w_0 = f(z_0)$, we can find an $U_0 \subset \Omega$ neighborhood of z_0, such that $f : U_0 \to W_0$ is injective (one-to-one function). Our proposed algorithms rely on a generalization of this statement to curves. Let

$$\Gamma := \{\gamma(s) : s \in J = [\alpha, \beta]\} \subset \Omega' \tag{2}$$

be a simple smooth curve with γ parameterization. That is, $\gamma : J \to \Gamma$ is a continuously differentiable bijection, for which $\gamma'(s) \neq 0$ $(s \in J)$. We say that the smooth function $\phi : J \to f^{-1}(\Gamma)$ is the inverse function of f along the curve Γ in notation

$$f(\phi) = \Gamma, \tag{3}$$

if, $f(\phi(s)) = \gamma(s)$ $(s \in J)$ holds. We will use the following theorem regarding the solutions of (3).

Theorem 1. *Suppose* $\Gamma \cap K' = \varnothing$. *Then,*

1. *For any* $z_0 \in f^{-1}(\Gamma)$, *equation* (3) *has a solution that passes through* z_0: $\exists \phi : \ \phi(s_0) = z_0, \ f(\phi(s)) = \gamma(s) \ (s \in J)$.
2. *If any two solutions of* (3) *have a common point, then these solutions coincide:* $\phi_1(s_0) = \phi_2(s_0) \implies \phi_1(s) = \phi_2(s) \ (s \in J)$.

A proof of Theorem 1 can be found in [5]. This relies on the existence and uniqueness theorem for differential equations. An alternative way to prove the theorem can be found in [11] (2.1, pp. 25).

If f is a rational function, then it has a finite number of critical points. Furthermore, supposing $\Gamma \cap K' = \varnothing$, the equation $f(\phi) = \Gamma$ has a finite number of $\phi_1, \ldots, \phi_m, \ m \in \mathbb{N}$ solutions. Based on Theorem 1, the ranges $\Gamma^j := \phi_j(J) \subset f^{-1}(\Gamma) \ (j = 1, \ldots, m)$ of these solutions, are distinct smooth curves.

3 An inverse algorithm for rational functions

In this section we discuss how to find the inverse curves $\Gamma^j \ (j = 1, \ldots, m)$. We propose a generalization of the inverse algorithm introduced in [5], where f was assumed to be a finite Blaschke-product:

$$B(z) := \varepsilon \prod_{k=1}^{m} \frac{z - a_k}{1 - \overline{a_k} z} \quad (z \in \overline{\mathbb{D}}, \ a_k \in \mathbb{D}, \ k = 1, \ldots, m, \ m \in \mathbb{N}, \varepsilon \in \mathbb{T}). \tag{4}$$

These special rational functions, as defined in (4), have many applications such as the construction of rational orthogonal systems [6]. Our algorithm proposed in [5] had two main ideas. First, we showed that if f is an m-factor Blaschke-product and we choose a point $w \in \mathbb{T}$, then every solution $z_i \in \mathbb{T}$ of

$$f(z_i) = w \ (i = 1, \ldots, m) \tag{5}$$

can easily be identified. In this work, we introduce a class of rational functions $\mathcal{R}$ and generalize this idea for $f \in \mathcal{R}$ in Section 3.3. The second idea of the algorithm in [5] was that given the initial solutions in (5), a successive application of Newton's iteration can be used to produce every inverse of the Blaschke-product f along the curve $\Gamma \subset \mathbb{D} \cup \mathbb{T}$.

The main contribution of this paper therefore is a generalization of the inverse algorithm introduced in [5] for a wide class of rational functions. We begin by comparing the proposed algorithm's properties to well-established root finding methods in Section 3.1. We will discuss a generalization of this iterative method for arbitrary analytic functions in Section 3.2. Furthermore, we are going to propose a method to identify every zero of the rational function $f \in \mathcal{R}$ in Section 3.4. Finally, we will investigate an alternative root finding algorithm involving the construction of Blaschke-products in Section 3.5.

3.1 Comparison with existing approaches

If f is a rational function, then for any $w \in \mathbb{C}$, the implicit equation

$$f(z) = w \tag{6}$$

can be rewritten as the polynomial root finding problem

$$P(z) = H(z) - w \cdot Q(z) = 0, \tag{7}$$

where H and Q are polynomials such that $f(z) = \frac{H(z)}{Q(z)}$. Many well-established numerical algorithms exist for solving such problems. In this section we will compare the proposed method to the well-known algorithms [1, 3]. Comparison with these methods makes sense, because both the proposed inverse algorithm for rational functions and [1, 3] were created to produce *every solution* of (6) and (7) simultaneously.

The Graeffe-Dandelin-Lobachesky method detailed in [3] introduces an iteration which squares the zeros of a polynomial in each step. This separates the roots by magnitude, then the Vieta-relations can be exploited to get good estimates on the absolute values of the roots. These estimates can either serve as a starting point for some other root finding algorithm, or one of numerous strategies can be applied to estimate the angles of the zeros as well.

Another well-known and popular algorithm for finding every zero of a polynomial is Aberth's method [1]. This algorithm is cubically convergent for simple zeros and can be interpreted as an improvement of the Durand-Kerner method [10]. Aberth's method updates an initial estimate of the roots in each step of the iteration. The iteration can encounter problems in the case when both the zeros of the polynomial and the initial approximations are distributed in a symmetrical fashion.

The advantages of the rational inverse algorithm proposed in this manuscript over the above mentioned well-known polynomial root finding methods are twofold. First, in order to acquire the form (7) from (6), one assumes that the values of the polynomials H and Q can be accessed separately. If the value of f is available in a sufficient number of points, one could apply interpolation to achieve this, however at the cost of possibly introducing numerical errors (especially in real life applications in the presence of noise). The second advantage of the proposed method is that it makes no assumptions on the order of f. The root finding algorithms [1, 3] require us to have apriori information about the order of the polynomial whose roots we are trying to identify. In contrast, the algorithm presented here can produce every solution to (6), regardless of the number of solutions, provided that f belongs to a certain class of rational functions. For some applications however this condition is naturally satisfied. For example our algorithm could presumably be applied to identify the zeros (and thus poles) of the transfer function of an all-pass filter [2] without knowing the order of the transfer function.

Finally, we would like to mention that our approach in considering rational functions for inverse problems instead of polynomials is not without precedent. In

fact, the classical Bernoulli-method [10] constructs a special rational function and identifies its so-called dominant pole in order to determine a zero of a polynomial.

3.2 Finding the inverses given an initial solution

Henceforth, denote by $D_r(z_0) := \{z \in \mathbb{C} : |z - z_0| < r\}$ and $\overline{D}_r(z_0) := \{z \in \mathbb{C} : |z - z_0| \le r\}$ the open and closed neighborhoods of z_0 and let $\Omega = D_R(0)$. Suppose the function f is analytic on $\overline{\Omega} = \overline{D}_R(0)$. Furthermore, let

$$M_j := \max_{z \in \overline{\Omega}} |f^{(j)}(z)|. \tag{8}$$

In order to produce the inverse curves $\Gamma^j \subset \Omega$ $(j = 1, \ldots, m)$ introduced in Section 2, we need to find neighborhoods which separate the curve $\Gamma \subset \Omega' = f(\Omega)$ from K' and the Γ^j curves from each other. Let

$$\rho(H, L) := \inf\{|z - w| : z \in H, w \in L\} \tag{9}$$

denote the distance between sets $H, L \subset \mathbb{C}$ and

$$\Gamma_r := \{w \in \Omega' : \rho(w, \Gamma) < r\} \tag{10}$$

denote the neighborhood of the curve Γ with a radius of r. In addition, let $K_r^c = \overline{\Omega} \setminus \cup_{\kappa \in K} D_r(\kappa)$ be the complement of the r radius neighborhood of the critical points. If $\Gamma \cap K' = \varnothing$, then Γ can be separated from K' in the following sense. There exists a number $r_1 > 0$ such that

$$\rho(\Gamma_{r_1}, K') > r_1. \tag{11}$$

By (11),

$$\rho(L, K) \ge \sqrt{r_1/M_2} =: r_2, \tag{12}$$

where $L := f^{-1}(\Gamma_{r_1})$. Indeed, if $\kappa \in K$, $w = f(z) \in \Gamma_{r_1}$, then

$$|f(z) - f(\kappa)| = |f(z) - f(\kappa) - f'(\kappa)(z - \kappa)| \le M_2 |z - \kappa|^2.$$

From here, (12) is a consequence of

$$\rho(\Gamma_{r_1}, K') \le M_2 \rho^2(K, L).$$

Since by Theorem 1, the inverse curves Γ^j are pairwise distinct, there exists r_0 for which

$$\Gamma_{r_0}^j \subset L, \ \Gamma_{r_0}^j \cap \Gamma_{r_0}^k = \varnothing, \quad j \ne k, \ 1 \le j, k \le m, \ L := f^{-1}(\Gamma_{r_1}). \tag{13}$$

Furthermore let

$$m_1 := \max_{z \in K_r^c} |1/f'(z)| \ge \max_{z \in L} |1/f'(z)|. \tag{14}$$

We note, that the constants m_1 and M_j only depend on Γ and f.

In order to solve the equation $f(\phi) = \Gamma$, suppose we already acquired for some $w_0 = \gamma(s_0) \in \Gamma$ point the solutions $z_{0,j} = \phi_j(s_0) \in \Gamma^j$ $(j = 1, \dots, m)$. We are going to discuss iterative methods, with which we can determine the inverses $z \in \Gamma^j$ of $w \in \Gamma$, provided w is close enough to w_0. The solution $z \in \Gamma^j$ can be found on the disk $\overline{D}_r(z_{0,j})$, as the fixed point of an iteration generated by the function

$$h(v) = v - (f(z_0 + v) - w) \cdot g(z_0 + v) \ (|v| < r). \tag{15}$$

Indeed, if g does not vanish, then

$$h(v) = v \iff f(z) = w \ (z := z_0 + v), \tag{16}$$

and since $|z - z_0| = |v| < r$, based on (13), z falls on Γ^j, provided $r < r_0$.

In order to find v which satisfies (16), we are going to show for some functions h, that they are contraction mappings. That is, for any $|v_k| < r$ $(k = 1, 2)$,

$$|h(v_1) - h(v_2)| < q \cdot |v_1 - v_2| \tag{17}$$

for some constant $q \in [0, 1)$. In Section 4, we provide specific examples of h and show that there exist $0 < r \leq r_0$ and $0 < \overline{r} < r_1$, such that

$$z_0 \in \Gamma^j, \ f(z_0) = w_0, \ w \in \Gamma, \ |w - w_0| \leq \overline{r} \implies h : \overline{D}_r \to \overline{D}_r \tag{18}$$

and h also possesses the property described in (17). Such mappings h satisfy the conditions of the Fixed-point theorem and therefore iterations of the type $v_{k+1} := h(v_k)$ will converge to the solution (16). Using these iterations, we can invert the function f in the $w_k := \gamma(s_k) \in \Gamma$ points, where s_k belongs to the partitioning $s_0 = \alpha < s_1 < \ldots < s_N = \beta \quad (J = [\alpha, \beta])$. If the partitioning is dense enough, beginning from some initial solution $z_0 \in \Gamma^j$ satisfying $f(z_0) = w_0 \in \Gamma$, we can find the rest of the solutions $z_k \in \Gamma^j$ for which $f(z_k) = w_k, \ (k = 1, \dots, N)$ recursively. These z_k solutions are the limits of fixed point iterations.

3.3 Finding an initial solution

In this section we introduce an algorithm to produce every initial solution $z_{0,j} = \phi(s_0) \in \Gamma^j, \ (j = 1, \dots, m)$. The proposed algorithm is a generalization of the method introduced in [5], where similar ideas were used to produce these solutions if f is an m-factor Blaschke-product (4).

We begin by specifying the class of rational functions $\mathcal{R}$, for which the discussed ideas are applicable. For a rational function f, let Z_f and P_f denote the set of its zeros and poles respectively. Let $\mathcal{R}$ be the class of rational functions, for which

$$R^* := \max\{|\xi| : \xi \in Z_f\} < R_* := \min\{|\zeta| : \zeta \in P_f\}. \tag{19}$$

Polynomials and Blaschke-products obviously belong to $\mathcal{R}$. We will make use of the notion of the Nyquist-plot, which for a function f belonging to $\mathcal{R}$ can be defined by (20).

$$\begin{gathered} f_{T_R} := f(T_R) \\ (T_R := \{z = R \cdot e^{it}, t \in \mathbb{I} = [-\pi, \pi)\},\ R^* < R < R_*,\ f \in \mathcal{R}). \end{gathered} \tag{20}$$

Our reason for considering the class of functions $\mathcal{R}$ is summarized by the next theorem.

Theorem 2. *If $f \in \mathcal{R}$, then the Nyquist-plot f_{T_R} can be written in the form*

$$f(Re^{it}) = A(t)e^{i\theta(t)}\ (t \in \mathbb{R}),$$

where A is a positive continuous function and $\theta : \mathbb{R} \to \mathbb{R}$ is a strictly increasing function. Furthermore θ satisfies $\theta(t + 2\pi) = \theta(t) + 2m\pi$ $(t \in \mathbb{R})$, where m denotes the number of f's zeros with multiplicities.

Proof. The winding number

$$\mathrm{Ind}(u, f_{T_R}) = \frac{1}{2\pi i} \int_{f_{T_R}} \frac{1}{z - u} dz\ (u \in \mathbb{C})$$

specifies the integer number of times the Nyquist-plot travels around the point u in a counter clockwise manner [7, 9, 11]. Cauchy's argument principle [7, 9, 11], makes a connection between the poles and zeros of f and the winding number of the Nyquist-plot at $u = 0$:

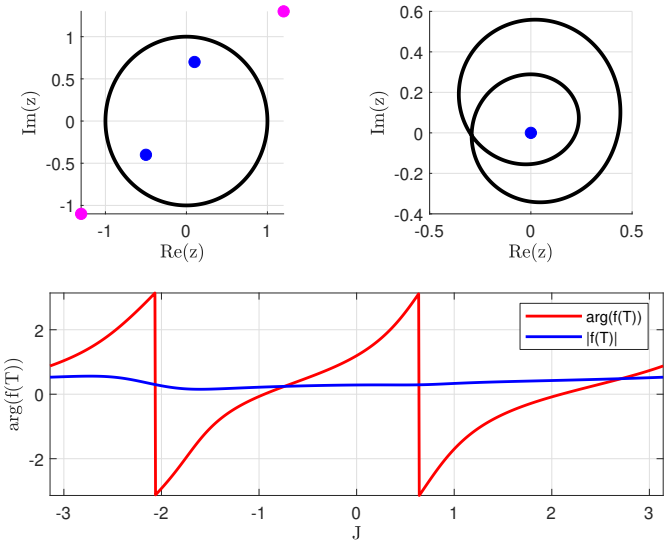
$$\mathrm{Ind}(0, f_{T_R}) = Z_{f,T_R} - P_{f,T_R},$$

where Z_{f,T_R} and P_{f,T_R} denote the number of zeros and poles that fall inside T_R. From this and the above mentioned interpretation of the winding number, choosing $f \in \mathcal{R}$ guarantees that in the Nyquist-plot

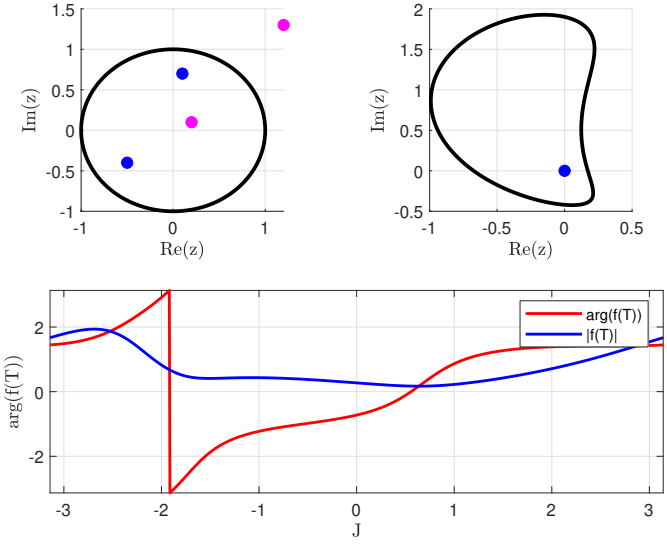
$$f(R \cdot e^{it}) = A(t)e^{i\theta(t)}\ (t \in \mathbb{R})$$

the argument function $\theta : \mathbb{R} \to \mathbb{R}$ is strictly increasing and satisfies $\theta(t + 2\pi) = \theta(t) + 2m\pi$ $(t \in \mathbb{R})$. □

We note that for Blaschke-products (4) $A(t) = 1$ $(t \in \mathbb{R})$. Figure 1 illustrates the Nyquist-plots of some examples of rational functions.



(a) The curve T_R with $R = 1$ (top left), its image $f(T_R)$ with respect to $f \in \mathcal{R}$ (top right), $\theta(t) \mod 2\pi$ and $A(t)$ (bottom).



(b) The curve T_R with $R = 1$ (top left), its image $f(T_R)$ with respect to $f \notin \mathcal{R}$ (top right), $\theta(t) \mod 2\pi$ and $A(t)$ (bottom).

Figure 1: 1a: Nyquist-plot of a rational function belonging to $\mathcal{R}$. The argument function θ is made up of 2 strictly increasing parts. 1b: Nyquist-plot of a rational function not in $\mathcal{R}$. Now the winding number is 1 and the Nyquist-plot makes a single revolution around 0. Blue points denote the zeros (and their images) of the functions, magenta points denote the poles.

From Theorem 2 it follows, that if $f \in \mathcal{R}$, each set

$$\mathbb{I}_\tau := \{t = \theta^{-1}(\theta(\tau) + 2j\pi) : t \in [-\pi, \pi),\ j \in \mathbb{Z}\} \quad (\tau \in [-\pi, \pi)) \tag{21}$$

has exactly m members. Furthermore, for any fixed τ we can easily produce the set $\mathbb{I}_\tau$ numerically (i.e. by interval halving). Then, we can identify m initial solutions by

$$f(Re^{it_j}) = f(z_{0,j}) = A(t_j)e^{i\theta(\tau)} = w_{0,j}\ (t_j \in \mathbb{I}_\tau,\ j = 1, \ldots, m). \tag{22}$$

3.4 Identifying every zero

In this section we discuss an application of the proposed inverse algorithm to find every zero of $f \in \mathcal{R}$. Then, we can give a parametric representation of the boundary of the star-like domain $f(D_R)$ as

$$F_R = \{A^*(\tau)e^{i\theta(\tau)} : \tau \in [-\pi, \pi)\}, \tag{23}$$

where $A^*(\tau) := \max_{t \in \mathbb{I}_\tau} A(t)$. The point $w \in f(D_R)$ is said to be an internal self intersecting point of the diagram $f(T_R)$, if there exist $t_1, t_2 \in \mathbb{I},\ t_1 \neq t_2$ that satisfy $f(Re^{it_1}) = f(Re^{it_2}) = w$. If $f \in \mathcal{R}$, then the S set of internal self intersecting points is finite. In order to find the zeros of f, we are going to produce the inverses along the line segments

$$\Gamma := [0, w_{0,j}] = \{\gamma(s) := (1 - s)w_{0,j} : 0 \leq s \leq 1\}, \tag{24}$$

that connect 0 with the initial points $w_{0,j}$ $(j = 1, \ldots, m)$. We only consider the inverses along the line segments for which

$$[0, w_{0,j}] \cap (K' \cup S) = \varnothing \tag{25}$$

holds. Let F_R^* denote the set of possible $w_{0,j}$ endpoints, with which the segment $[0, w_{0,j}]$ satisfies (25). The set $F_R \setminus F_R^*$ is a finite set. We bring attention to the fact, that if the initial inverse points were determined according to Section 3.3, then the points $w_{0,j}$ all fall on the same line segment $(j = 1, \ldots, m)$. Henceforth we assume that the elements of $\mathbb{I}_\tau$ are indexed in a way so that the points $w_{0,j} = A(t_j)e^{i\theta(\tau)}$ satisfy $|w_{0,1}| < |w_{0,2}| < \ldots < |w_{0,m}|$ and therefore

$$[0, w_{0,1}] \subset [0, w_{0,2}] \subset \ldots \subset [0, w_{0,m}]. \tag{26}$$

Suppose that f has only simple roots. Consider the functions $\phi_j : [0, A(t_j)] \to f^{-1}(\Gamma)$ starting from the origin going backwards. That is, as a first step we define the inverse images of the segment $[0, w_{0,1}]$, which start from the m zeros of f. Now $\phi(A(t_1)) = z_{0,1} = Re^{it_1}$. Taking the inverse images of the segment $[w_{0,1}, w_{0,2}]$ starting from the point $\phi_j(t_1)$ $(j = 2, \ldots, m)$, we get $m - 1$ smooth curves, furthermore $\phi_1(A(t_2)) = z_{0,2} = Re^{it_2}$. Continuing this method finally brings us to consider the inverse of $[w_{0,m-1}, w_{0,m}]$ starting from the point $\phi_m(t_{m-1})$, which gives

us a smooth curve ending in $\phi_1(A(t_m)) = z_{0,m} = Re^{it_m}$. Thus, we showed that the functions ϕ_j considered over the intervals $[0, A(t_j)]$ are smooth solutions of the equation $f(\phi) = [0, w_{0,j}]$. Furthermore, these solutions connect the $z_{0,j}$ points on the boundary with the zeros of f. Our numerical experiments show, that if a zero of f has a multiplicity greater than 1, then the number of ϕ_j solution trajectories ending in this root matches the multiplicity. Figures 2 and 3 illustrate the above described root finding algorithm.

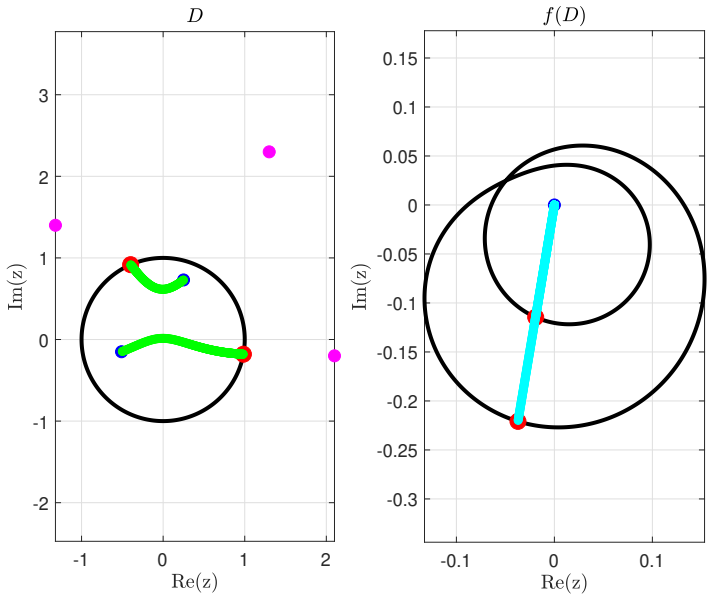
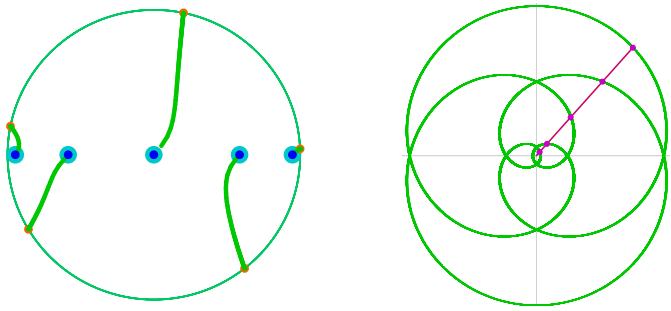


Figure 2: LEFT: The domain D_R (bordered by black circle), the zeros (blue points) and the poles (purple points) of $f \in \mathcal{R}$, initial inverse points (red points on the circle), and the inverse curves Γ^1 and Γ^2 (green curves). RIGHT: The range $f(D_R)$ bordered by the Nyquist-plot (black curve), $w_{0,1}$ and $w_{0,2}$ (red points), the inverted line segments $[0, w_{0,1}]$ and $[0, w_{0,2}]$ (light blue segments)

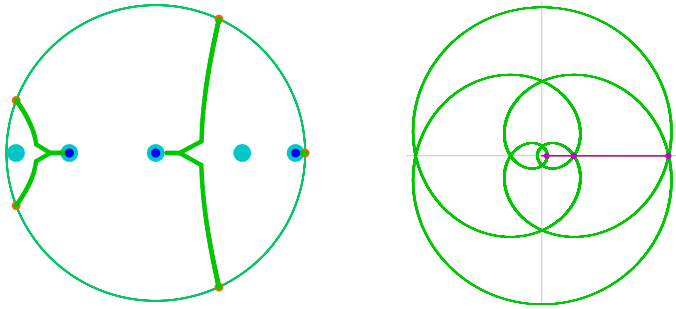
3.5 Construction of equivalent Blaschke-products

We now detail an alternative approach to identify the zeros of polynomials. Namely, we will construct Blaschke-products (4), whose zeros match the zeros the polynomial in question, then apply the inverse algorithm introduced in [5] to identify these. Suppose first that P is a polynomial of degree m. We can then consider the reciprocate polynomial P_r:

$$P_r(z) := z^m \overline{P}(1/\overline{z}) \ (z \in \mathbb{C}). \tag{27}$$



(a) LEFT: The complex unit circle, containing the roots (blue points) of a Chebyshev-polynomial. The inverse curves found by the proposed inverse algorithm are colored green. RIGHT: The Nyquist-plot $f(T_R)$ and the line segments $[0, w_{0,j}]$ $(j = 1, \ldots, 5)$ to be inverted. Here, $[0, w_{0,j}] \cap S = \varnothing$.



(b) LEFT: The complex unit circle, containing the roots (blue points) of a Chebyshev-polynomial. The inverse curves found by the proposed inverse algorithm are colored green. RIGHT: The Nyquist-plot $f(T_R)$ and the line segments $[0, w_{0,j}]$ $(j = 1, \ldots, 5)$ to be inverted. Here, $[, w_{0,j}] \cap S \neq \varnothing$.

Figure 3: Internal self intersecting points S

Using (27), we can construct the m-factor Blaschke-product B:

$$B(z) := \frac{P(z)}{P_r(z)} = \prod_{i=k}^{m} \frac{z - a_k}{1 - z\overline{a}_k}, \tag{28}$$

where a_k, $(k = 1, \ldots, m)$ are the zeros of P including multiplicities. Then, the algorithm described in [5] can be applied to find the zeros a_k, $(k = 1, \ldots, m)$.

4 Fixed point iterations

In this section we are going to give some concrete examples for the contraction mappings (15) and consider their properties. More precisely, we suppose that for a rational function $f \in \mathcal{R}$, we already have an initial inverse point z_0 satisfying $f(z_0) = w_0$. We are going to show that the proposed iterations satisfy (18) and (16),

hence they produce the inverse at a point $w \in \Gamma$ as explained in 3.2, provided w is close enough to w_0.

4.1 A linearly convergent iteration

Our first example is a linearly convergent iterative method. Let

$$h(v) := v - \frac{f(z_0 + v) - w}{f'(z_0)} \quad (|v| < r), \tag{29}$$

where $0 < r$ is assumed to satisfy $r < r_0$, in accordance with (18) and (13). We are going to show that there exists $r_1 > \overline{r} > 0$ such that if $|w - w_0| < \overline{r}$ and r is sufficiently small, then $h : \overline{D}_r \to \overline{D}_r$ is a contraction mapping. Then, according to Section 3.2, for the limit

$$v^* = \lim_{k\to\infty} v_k, \ v_{k+1} := h(v_k), \ v_0 = 0 \tag{30}$$

$f(v^* + z_0) = w$ will hold. Notice, that h has the following properties:

1. $h'(0) = 0$,
2. $|h''(v)| \le M_2 \cdot m_1 \quad (|v| \le r)$,

where M_2 and m_1 only depend on f and Γ as defined in (14) and (8). Choosing $v_1, v_2 \in \overline{D}_r$, we get:

$$|h(v_1) - h(v_2)| \le \max_{s\in[v_1,v_2]} |h'(s)| \cdot |v_1 - v_2| \le M_2 \cdot r \cdot |v_1 - v_2|. \tag{31}$$

Furthermore, if $v \in \overline{D}_r$

$$|h(v)| \le |h(v) - h(0)| + |h(0)| \le M_2 \cdot r \cdot |v| + |w - w_0| \cdot m_1. \tag{32}$$

From (31) and (32), choosing $r := \min\{r_0, 1/(2M_2)\}$ and w such that $|w - w_0| = \min\{\frac{r}{2m_1}, r_1\}$ hold guarantees that $h : \overline{D}_r \to \overline{D}_r$ is a contraction:

$$|h(v_1) - h(v_2)| \le \frac{1}{2}|v_1 - v_2|, \ |h(v)| \le r \quad (v, v_1, v_2 \in \overline{D}_r).$$

Convergence of (30) then follows from the Fixed-point theorem and the inverse property $h(v) = v \iff f(z_0 + v) = w$ is guaranteed by the considerations in 3.2. We can also apply the Fixed-point theorem to get the error estimate

$$|v_n - v^*| \le 2^{-n+1} \ (n \in \mathbb{N}). \tag{33}$$

We note that a slight modification of h yields the iteration

$$\tilde{h}(v) := v - \frac{f(z_0 + v) - w}{f'(z_0 + v)} \quad (|v| \le r \le r_0) \tag{34}$$

which shows locally quadratic convergence. The iterative method generated by (34) can be interpreted as a Newton-iteration aimed at finding a zero of the function $g(v) := f(z_0 + v) - w_0$.

4.2 Identifying the zeros of Blaschke-products without access to derivatives

Suppose we are trying to identify the zeros of $f \in \mathcal{R}$, where f is an m-factor Blaschke-product (4). Suppose furthermore that every solution of $f(z_{0,k}) = w_0$ $(k = 1, \ldots, m)$ has already been acquired for some w_0. In this section we are going to construct a polynomial P based on the initial solutions $z_{0,k}$, whose zeros match the zeros of f. We can then apply the generalized inverse algorithm proposed in this paper to identify the zeros of P, thus identifying the zeros and poles of f. In addition, we are going to show, that when solving the implicit problems $P(z) = w_j$ $(j > 0)$, we can express the derivative $P'(z_0)$ in (29) using the solutions from previous steps of the algorithm. This in turn means that when f is an m-factor Blaschke-product, one can find all of its zeros using the proposed inverse algorithm with a variation of the iteration (29), where we can express the needed derivative values from previous solutions.

In Section 3.3 we saw that if we choose the right side of $f(z) = w_0$ carefully, then every solution can be found with a simple numerical method (i.e. by interval halving). If, for example the disk D_R contains every zero of a polynomial H, then for any $w_0 = H(Re^{it_0})$ value, every zero of $Q(z) := H(z) - w_0$ can be easily identified. These could be used as the initial solutions for the proposed algorithm in 3.2. We can, however have other uses for the $z_1, \ldots, z_m$ (pairwise different) zeros of Q as well. Namely, since $H' = Q'$, we can use the $Q(z) = q_m \cdot \prod_{k=1}^{m}(z - z_k)$ form of Q to calculate the derivatives of H. Here q_m denotes the leading coefficient of Q. Provided we have access to q_m, we can easily construct the derivative values needed for the linearly convergent iteration (29) using the initial solutions.

We are going to extend this idea to m-factor Blaschke-products. Namely, we are going to construct an m degree polynomial P with a leading coefficient $p_m = 1$, whose zeros match the zeros of the Blaschke-product. Then, the proposed inverse algorithm and the above idea can be used to identify these zeros. Suppose f is a Blaschke-product and for some $w_0 \in \mathbb{T}$, all m solutions to $f(z) = w_0$ have already been found. Since

$$f(z) = \prod_{k=1}^{m} \frac{z - a_k}{1 - \overline{a}_k z} \ (a_k \in \mathbb{D}, z \in \mathbb{D} \cup \mathbb{T}),$$

the solutions $z_1, \ldots z_m$ coincide with the roots of the m degree polynomial

$$P_{w_0}(z) := \prod_{k=1}^{m}(z - a_k) - w_0 \cdot \prod_{k=1}^{m}(1 - \overline{a}_k z) = \sum_{k=0}^{m} p_{w_0,k} \cdot z^k. \tag{35}$$

From (35), the leading coefficient is $p_{w_0,m} = 1 - w_0 \cdot (-1)^m \prod_{k=1}^{m} \overline{a}_k$. Notice, that since f was a Blaschke-product, the leading coefficient can also be written as $p_{w_0,m} = 1 - w_0 \cdot \overline{f(0)}$. This means, we can write the polynomial P_{w_0} using the solutions $z_1, \ldots, z_m$ to $f(z) = w_0$ as

$$P_{w_0}(z) = \left(1 - w_0 \cdot \overline{f(0)}\right) \prod_{k=1}^{m} (z - z_k). \tag{36}$$

Now consider the equation $f(u) = -w_0$. If $w_0 \in \mathbb{T}$, then the ideas discussed in 3.3 can be used to identify all m solutions to this. These $u_1, \ldots, u_m$ solutions are also the zeros of the polynomial

$$P_{-w_0}(u) = \prod_{k=1}^{m} (u - a_k) + w_0 \cdot \prod_{k=1}^{m} (1 - \overline{a}_k u), \tag{37}$$

which by the above can be written using the solutions to $f(u) = -w_0$ as

$$P_{-w_0}(z) = \left(1 + w_0 \cdot \overline{f(0)}\right) \prod_{k=1}^{m} (z - u_k). \tag{38}$$

By equations (36) and (38) we can query the values of P_{w_0} and P_{-w_0} using the solutions and $f(0)$, while by equations (35) and (37)

$$P(z) = \frac{1}{2} \left(P_{w_0}(z) + P_{-w_0}(z)\right) = \prod_{k=1}^{m} (z - a_k). \tag{39}$$

In (39), $P(z)$ is an m degree polynomial with a leading coefficient $p_m = 1$, whose zeros a_k $(k = 1, \ldots, m)$ match the zeros of the original Blaschke-product f.

4.3 Secant method

We now discuss an alternative to (29), where we replace the derivatives in (29) with divided differences. For $v \in \overline{D}_r$, let (15) take the form

$$h(v) := v - \frac{f(z_0 + v) - w}{f[z_0 + v, z_0]} = v - \frac{v \cdot (f(z_0 + v) - w)}{f(z_0 + v) - f(z_0)}, \ \lim_{v \to 0} h(v) = \frac{w - w_0}{f'(z_0)}, \tag{40}$$

where $f \in \mathcal{R}$, $f(z_0) = w_0 \in \Gamma$, $|w - w_0| < \overline{r}$ and $r < r_0$ in accordance with (18). We are going to show, that such r and $\overline{r}$ exist for (40). Then, by the ideas in 3.2 the limit

$$v^* = \lim_{k \to \infty} v_k, \ v_{k+1} := h(v_k), \ 0 < |v_0| < r$$

satsifies $f(v^* + z_0) = w$.

Consider the Taylor-series of f around z_0:

$$f(z_0 + v) = f(z_0) + f'(z_0) \cdot v + \sum_{k=2}^{\infty} \frac{f^{(k)}(z_0)}{k!} \cdot v^k. \tag{41}$$

Equation (41) gives us

$$f[z_0+v,z_0]=f'(z_0)+\varepsilon(v)=f'(z_0)+v\cdot\varepsilon_1(v),$$

where $\varepsilon_1(v)=\sum_{k=2}^{\infty}\frac{f^{(k)}(z_0)}{k!}v^{k-2}$. Notice that

$$\varepsilon(0)=0,\quad \varepsilon'(0)=\frac{f''(z_0)}{2} \tag{42}$$

hold.

Since

$$h(v):=v-\frac{f(z_0+v)-w}{f'(z_0)+\varepsilon(v)}=v-f_1(v)\cdot g(v), \tag{43}$$

where $f_1(v)=f(z_0+v)-w$ and $g(v)=1/(f'(z_0)+\varepsilon(v))$, we can write the second derivative function of h as

$$h''=f_1''g+2f_1'g'+f_1g''=f_1''g-2f_1'\varepsilon'g^2+f_1(-\varepsilon''g^2+2\varepsilon'^2g^3).$$

The derivatives $f_1^{(j)}$, $\varepsilon^{(j)}$ $(j\le 2)$ are bounded on $\overline{\Omega}$. We are going to show, that for sufficiently small r, the function $1/g$ is bounded from below on Γ^j. Indeed, for $z_0\in\Gamma^j$,

$$1/|g(v)|\ge|f'(z_0)|-|v||\varepsilon_1(v)|\ge 1/m_1-|v|m_2, \tag{44}$$

where $m_2:=\max_{|v|<r}|\varepsilon_1(v)|$ and m_1 is defined in (14). From this, if $|v|\le r:=\frac{1}{2m_2m_1}$, then $|g(v)|\le 2m_1$. It follows that h'' is bounded from above:

$$|h''(v)|\le m_3\ (|v|\le r). \tag{45}$$

In order to show that h is a contraction mapping, we introduce the function

$$h_1(v)=h(v)-v\cdot h'(0). \tag{46}$$

It is clear, that for h_1,

$$h_1'(0)=0 \tag{47}$$

holds. Using (47) and the mean value theorem, we get that for any $v_1,v_2\in\overline{D}_r$:

$$|h_1(v_1)-h_1(v_2)|\le\max_{v\in[v_1,v_2]}|h_1'(v)||v_1-v_2|\le m_3\cdot r\cdot|v_1-v_2|. \tag{48}$$

Now we can use (48) to show h is a contraction mapping. Let $v_1,v_2\in\overline{D}_r$, then

$$\begin{aligned}&|h(v_1)-h(v_2)|=|(h_1(v_1)+h'(0)v_1)-(h_1(v_2)+h'(0)v_2)|\le\\&m_3\cdot r\cdot|v_1-v_2|+|h'(0)||v_1-v_2|=(m_3\cdot r+|h'(0)|)|v_1-v_2|\end{aligned} \tag{49}$$

and

$$|h(v)|\le|h(v)-h(0)|+|h(0)|\le(m_3r+|h'(0)|)|v|+|h(0)|. \tag{50}$$

If we now choose

$$r \leq \min\{r_0, 1/(4m_3)\}$$
$$|w - w_0| \leq \overline{r} \leq \min\{r_1, 1/(2m_1^2 M_2), r/(2m_1)\},$$

then by (40) and (43)

$$|h(0)| \leq |w - w_0| m_1 \leq r/2, \quad |h'(0)| \leq |w - w_0| M_2 m_1^2/2 \leq r/4 \tag{51}$$

and consequently

$$|h(v_1) - h(v_2)| \leq |v_1 - v_2|/2, \quad |h(v)| \leq |v|/2 + r/2 \leq r \ (|v| \leq r). \tag{52}$$

Equation (52) shows that $h : \overline{D}_r \to \overline{D}_r$ is a contraction mapping. Thus, by the ideas in 3.2 and the fixed point theorem, the iteration generated by (40) can be used to find the inverse of f at a suitable point w.

5 Conclusion

In this study, we examined the numerical construction of the inverseses of rational functions along a curve. We considered the existence of continous solution curves in Section 2. We then provided an iterative algorithm to produce these solutions numerically in Section 3.2, given some initial solution points. We also proposed a class of rational functions, for which we can easily identify the needed initial solutions in Section 3.3. Furthemore, we proposed an algorithm with which the inverses can be used to identify the zeros of rational functions in Section 3.4. We gave an alternative algorithm for root finding in the case, when f is a polynomial, whose main feature was the construction of special Blaschke-products in Section 3.5. Finally, we investigated fixed point iterations to be used with our iterative algorithm and proved their convergence properties in Section 4.

The investigated algorithms give rise to a number of interesting applications, such as the identification of transfer functions for SISO (single input, single output) systems. We plan to explore these applications in future works.

References

[1] Aberth, Oliver. Iteration methods for finding all zeros of a polynomial simultaneously. *Mathematics of Computation*, 27(122):339–344, 1973. DOI: 10.1090/S0025-5718-1973-0329236-7.

[2] Appaiah, Kumar and Pal, Debasattam. All-pass filter design using Blaschke interpolation. *IEEE Signal Processing Letters*, 27:226–230, 2020. DOI: 10.1109/LSP.2020.2965318.

[3] Best, G. C. Notes on the Graeffe method of root squaring. *The American Mathematical Monthly*, 56(2):91–94, 1949. DOI: 10.2307/2306166.

[4] Dozsa, T. and Schipp, F. Hyperbolic geometry and Blaschke-functions. *Annales Univ. Sci. Budapest. Sect. Comp.*, 51:59–68, 2020. URL: http://ac.inf.elte.hu/Vol_051_2020/059_51.pdf.

[5] Dozsa, T. and Schipp, F. A generalization of the root function. *Annales Univ. Sci. Budapest. Sect. Comp.*, 52:97–108, 2021. URL: http://ac.inf.elte.hu/Vol_052_2021/097_52.pdf.

[6] Fridli, S. and Schipp, F. Discrete rational biorthogonal systems on the disc. *Annales Univ. Sci. Budapest. Sect. Comp.*, 50:127–134, 2020. URL: http://ac.inf.elte.hu/Vol_050_2020/127_50.pdf.

[7] Henrici, P. *Applied and Computational Complex Analysis, Volume 3: Discrete Fourier Analysis, Cauchy Integrals, Construction of Conformal Maps, Univalent Functions*. Wiley Classics Library. Wiley, 1993.

[8] Kovács, Péter, Fridli, Sándor, and Schipp, Ferenc. Generalized rational variable projection with application in ECG compression. *IEEE Transactions on Signal Processing*, 68:478–492, 2020. DOI: 10.1109/TSP.2019.2961234.

[9] Rudin, W. *Real and Complex Analysis*. Higher Mathematics Series. McGraw-Hill Education, 1987.

[10] Sendov, Bl., Andreev, A., and Kjurkchiev, N. Numerical solution of polynomial equations. Volume 3 of *Handbook of Numerical Analysis*, pages 625–778. Elsevier, 1994. DOI: 10.1016/S1570-8659(05)80019-5.

[11] Sheil-Small, T. *Complex Polynomials*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2002. DOI: 10.1017/CBO9780511543074.

[12] Soumelidis, Alexandros, Bokor, József, and Schipp, Ferenc. Applying hyperbolic wavelet constructions in the identification of signals and systems. *IFAC Proceedings Volumes*, 42(10):1334–1339, 2009. DOI: 10.3182/20090706-3-FR-2004.00222.

[13] Tan, Chunyu, Zhang, Liming, and Wu, Hau-tieng. A novel Blaschke unwinding adaptive-Fourier-decomposition-based signal compression algorithm with application on ECG signals. *IEEE Journal of Biomedical and Health Informatics*, 23(2):672–682, 2019. DOI: 10.1109/JBHI.2018.2817192.

[14] Van den Hof, Paul, Wahlberg, Bo, Heuberger, Peter, Ninness, Brett, Bokor, Jozsef, and Oliveira e Silva, Tomás. Modelling and identification with rational orthogonal basis functions. *IFAC Proceedings Volumes*, 33(15):445–455, 2000. DOI: 10.1016/S1474-6670(17)39791-4.

Acta Cybernetica **26** (2023) 71–82.

On Some Convergence Properties for Finite Element Approximations to the Inverse of Linear Elliptic Operators*

Takehiko Kinoshita[a], Yoshitaka Watanabe[b], and Mitsuhiro T. Nakao[c]

Abstract

This paper deals with convergence theorems of the Galerkin finite element approximation for the second-order elliptic boundary value problems. Under some quite general settings, we show not only the pointwise convergence but also prove that the norm of approximate operator converges to the corresponding norm for the inverse of a linear elliptic operator. Since the approximate norm estimates of linearized inverse operator play an essential role in the numerical verification method of solutions for non-linear elliptic problems, our result is also important in terms of guaranteeing its validity. Furthermore, the present method can also be applied to more general elliptic problems, e.g., biharmonic problems and so on.

Keywords: linear elliptic problems, finite element approximation, norm estimation of the inverse operator, convergence theorem

1 Introduction

In this section, we describe the background of the present study with notations of related function spaces, including finite elements, and the formulation of the problem. We will also mention the previous results that motivated this article.

1.1 Notations

We now introduce some function spaces necessary to consider the concerned problems.

*This work was supported by Grants-in-Aid from the Ministry of Education, Culture, Sports, Science and Technology of Japan (Nos. 21H01000, 21K03373, 21K03378) and Japan Science and Technology Agency, CREST (No. JP-MJCR14D4).

[a]Department of Mathematical Science, Saga University, Saga 840-8502, Japan, E-mail: kinosita@cc.saga-u.ac.jp, ORCID: 0000-0001-9756-4571

[b]Research Institute for Information Technology, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan, ORCID: 0000-0001-6520-3552

[c]Faculty of Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan, ORCID: 0000-0001-5228-0591

DOI: 10.14232/actacyb.294906

Let $\Omega \subset \mathbb{R}^d$ be a bounded polygonal or polyhedral domain where $d \in \{1, 2, 3\}$. For a non-negative integer m, let $H^m(\Omega)$ be the real L^2 Sobolev space with order m on Ω. We define

$$H_0^1(\Omega) := \left\{ u \in H^1(\Omega) \,\middle|\, u = 0 \text{ on } \partial\Omega \right\}$$

then $H_0^1(\Omega)$ is a Hilbert space with respect to the inner product $(u, v)_{H_0^1(\Omega)} := (\nabla u, \nabla v)_{L^2(\Omega)^d}$ and its norm is given by $\|u\|_{H_0^1(\Omega)} := \sqrt{(u, u)_{H_0^1(\Omega)}}$ where $(\,\cdot\,,\,\cdot\,)_{L^2}$ is the usual L^2 inner product on Ω. Let $H^{-1}(\Omega)$ be the dual space of $H_0^1(\Omega)$.

For a given non-linear function $f : H_0^1(\Omega) \to H^{-1}(\Omega)$ with certain properties, we often consider the existence and local uniqueness of the solution u satisfying the following non-linear elliptic boundary value problem of the form (e.g. [6] etc.):

$$\begin{cases} -\triangle u = f(u) & \text{in } \Omega \quad (1a) \\ \quad u = 0 & \text{on } \partial\Omega. \quad (1b) \end{cases}$$

To prove the existence of the solution of (1a)-(1b), the information on the linearized operator $\mathscr{L} := -\triangle - f'(u_k) : H_0^1(\Omega) \to H^{-1}(\Omega)$ and its inverse play important roles where u_k is a suitable approximation of u and $f'(u_k)$ is the Fréchet derivative of f at u_k. Moreover, we assume that $f'(u_k) \in \mathcal{L}\big(H_0^1(\Omega), L^2(\Omega)\big)$ for u_k with suitable regularities and the weak Laplace operator $-\triangle \in \mathcal{L}\big(H_0^1(\Omega), H^{-1}(\Omega)\big)$ where $\mathcal{L}(X, Y)$ is the linear space of all bounded linear operators from X to Y. As well known, by the Riesz representation lemma, the Poisson equation with homogeneous Dirichlet boundary condition is uniquely solvable. Namely, there exists a bounded inverse operator of $-\triangle$ such that $(-\triangle)^{-1} \in \mathcal{L}\big(H^{-1}(\Omega), H_0^1(\Omega)\big)$. Then, $\mathscr{L}$ can be represented as $\mathscr{L} = (-\triangle)\big(I - (-\triangle)^{-1} f'(u_k)\big)$ where I is the identity map on $H_0^1(\Omega)$. We denote $A := (-\triangle)^{-1} f'(u_k) \in \mathcal{L}\big(H_0^1(\Omega)\big)$. Note that A is a compact operator on $H_0^1(\Omega)$.

For an arbitrary $w \in H_0^1(\Omega)$, we set $u := Aw \in H_0^1(\Omega)$. Then, u satisfies the following variational equation:

$$(\nabla u, \nabla v)_{L^2(\Omega)^d} = \big(I_e g\big)(v) \quad \forall v \in H_0^1(\Omega) \tag{2}$$

where $I_e : L^2(\Omega) \hookrightarrow H^{-1}(\Omega)$ is an embedding operator and $g := f'(u_k)w \in L^2(\Omega)$. By some standard arguments using the Riesz representation theorem, we can rewrite (2) simply as

$$(\nabla u, \nabla v)_{L^2(\Omega)^d} = (g, v)_{L^2(\Omega)} \quad \forall v \in H_0^1(\Omega). \tag{3}$$

In general, the regularity of the solution (3) is smoother than $H_0^1(\Omega)$. Particularly, $u \in H\big(\triangle; L^2(\Omega)\big)$ holds where $H\big(\triangle; L^2(\Omega)\big) := \left\{ u \in H_0^1(\Omega) \,\middle|\, \triangle u \in L^2(\Omega) \right\}$.

Note that, if there exists a bounded inverse of $I - A$, then $\mathscr{L}$ also has an inverse: $\mathscr{L}^{-1} = (I - A)^{-1}(-\triangle)^{-1}$, and that $\big\|\mathscr{L}^{-1}\big\|_{\mathcal{L}\left(H^{-1}(\Omega), H_0^1(\Omega)\right)} = \big\|(I - A)^{-1}\big\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}$ holds (also see [4, Remark 1.3]).

Nakao et al. [5, 7] proposed numerical verification approaches for computing upper bounds of $\left\|\mathscr{L}^{-1}\right\|_{\mathcal{L}\left(H^{-1}(\Omega),H_0^1(\Omega)\right)}$ (*cf.* [9, 10, 12, 3]).

Now, in order to define the approximation to the inverse operator $\mathscr{L}^{-1}$, we introduce the finite element space in the most general way possible. Let $S_h(\Omega)$ be a finite-dimensional subspace of $H_0^1(\Omega)$ depending on the discretization parameter $h > 0$ corresponding to the mesh size. We define the H_0^1-projection P_h from $H_0^1(\Omega)$ to $S_h(\Omega)$ such that

$$(u - P_h u, v_h)_{H_0^1(\Omega)} = 0 \quad \forall v_h \in S_h(\Omega). \tag{4}$$

Let $\{\phi_i\}_{i=1}^n \subset H_0^1(\Omega)$ be the set of basis functions in $S_h(\Omega)$ where $n := \dim S_h(\Omega)$. Let D_ϕ and G_ϕ be n-by-n matrices whose (i, j) elements are defined by

$$\begin{aligned} D_{\phi,i,j} &= (\nabla\phi_j, \nabla\phi_i)_{L^2(\Omega)^d}, \\ G_{\phi,i,j} &= (\nabla\phi_j, \nabla\phi_i)_{L^2(\Omega)^d} - (f'(u_k)\phi_j, \phi_i)_{L^2(\Omega)}, \end{aligned}$$

where matrix G_ϕ is the corresponding representation to the Galerkin approximation of operator $\mathscr{L}$. Since D_ϕ is a positive definite matrix, it can be Cholesky decomposed as $D_\phi = E_\phi E_\phi^T$ where E_ϕ is a lower triangular matrix and E_ϕ^T is the transposed matrix of E_ϕ. We define the Galerkin approximation of $I - A$ by $[I - A]_h := P_h(I - A)|_{S_h(\Omega)} : S_h(\Omega) \to S_h(\Omega)$ where $(I - A)|_{S_h(\Omega)}$ is the restriction of $I - A$ on $S_h(\Omega)$ and let $[I - A]_h^{-1} := \left(P_h(I - A)|_{S_h(\Omega)}\right)^{-1}$, if the inverse exists. Then, $\left\|[I - A]_h^{-1}\right\|_{\mathcal{L}\left(S_h(\Omega)\right)} = \left\|E_\phi^T G_\phi^{-1} E_\phi\right\|_2 =: r_h$ holds where $\|\cdot\|_2$ is the matrix 2-norm / the spectral matrix norm (see [5]). Since the non-singularity of the matrix can be verified by computational procedure (see, e.g., [11]), the existence of $[I - A]_h^{-1}$ is usually assumed to be valid([5]).

1.2 Motivation and preliminary results

In this subsection, we describe the previous results mainly obtained in [4], which is the motivation of this study.

Suppose that P_h defined by (4) has the following convergence property

$$\lim_{h\to 0} \|P_h u - u\|_{H_0^1(\Omega)} = 0, \quad \forall u \in H_0^1(\Omega) \tag{5}$$

and that there exists a positive constant $\tilde{C}(h)$ such that $\tilde{C}(h) \to 0$ as $h \to 0$ and satisfying

$$\|\nabla(u - P_h u)\|_{L^2(\Omega)^d} \le \tilde{C}(h) \|\triangle u\|_{L^2(\Omega)}, \quad \forall u \in H\left(\triangle; L^2(\Omega)\right). \tag{6}$$

The conditions (5) and (6) are satisfied for usual finite element subspaces (see, e.g., [1, 2, 8] etc.). Also, note that the following estimates hold for arbitrary $u \in H_0^1(\Omega)$:

$$\begin{aligned} \|(I - P_h)Au\|_{H_0^1(\Omega)} &\le \tilde{C}(h) \|\triangle Au\|_{L^2(\Omega)} \\ &\le \tilde{C}(h) \|f'(u_k)\|_{\mathcal{L}\left(H_0^1(\Omega),L^2(\Omega)\right)} \|u\|_{H_0^1(\Omega)}. \end{aligned} \tag{7}$$

We now suppose that the linearized operator $f'(u_k)$ is represented as $f'(u_k)u = -b \cdot \nabla u - cu$ for some functions such that $b \in W^{1,\infty}(\Omega)^d$ and $c \in L^\infty(\Omega)$. And we set the following non-negative constants:

$$\begin{aligned} C_1 &:= \|b\|_{L^\infty(\Omega)^d} + C_p \|c\|_{L^\infty(\Omega)}, \\ C_2 &:= \|b\|_{L^\infty(\Omega)^d} + \tilde{C}(h) \|c\|_{L^\infty(\Omega)}, \\ K(h) &:= \tilde{C}(h) \left(C_p \|\nabla \cdot b\|_{L^\infty(\Omega)} + C_1 \right) \end{aligned}$$

where C_p is the Poincaré constant satisfying

$$\|u\|_{L^2(\Omega)} \le C_p \|\nabla u\|_{L^2(\Omega)^d} \quad \forall u \in H_0^1(\Omega).$$

Then we already obtain the following existential condition and estimates of the linearized inverse operator $(I - A)^{-1}$:

Theorem 1 ([7, Theorem 2]). *If $\kappa_h := \tilde{C}(h)\big(r_h K(h) C_1 + C_2\big) < 1$, then $I - A$ is invertible and the following estimate holds:*

$$\left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} \le \frac{1}{1-\kappa_h} \left\| \begin{pmatrix} r_h\big(1 - C_2\tilde{C}(h)\big) & r_h K(h) \\ r_h C_1 \tilde{C}(h) & 1 \end{pmatrix} \right\|_2 .$$

Moreover, by using the above theorem, if $\{r_h\}_{h>0}$ is a convergent sequence, then we have

$$\begin{aligned} \left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} &\le \lim_{h\to 0} \frac{1}{1-\kappa_h} \left\| \begin{pmatrix} r_h\big(1 - C_2\tilde{C}(h)\big) & r_h K(h) \\ r_h C_1 \tilde{C}(h) & 1 \end{pmatrix} \right\|_2 \\ &= \left\| \begin{pmatrix} \lim\limits_{h\to 0} r_h & 0 \\ 0 & 1 \end{pmatrix} \right\|_2 \\ &= \max\left\{ \lim_{h\to 0} r_h, 1 \right\}. \end{aligned} \tag{8}$$

In our previous paper [4], by using (8), we presented the following relation:

$$1 \le \left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} \le \lim_{h\to 0} \left\|[I-A]_h^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}, \tag{9}$$

provided that the limit in (9) actually exists. However, the question remains whether the second inequality of (9) becomes equality. In this paper, we prove that such equality holds true as well as clarify the condition for the existence of $[I - A]_h^{-1}$.

2 Main results

In this section, based on the notations and the preliminaries introduced in previous sections, we present the main result on the convergence property for finite element

approximations of an inverse elliptic operator. To proceed with the argument, in the following, although it may be duplicated, some new definitions and assumptions are made again. It should also be noted that the intended purpose is achieved under a very common setting of the finite element space and approximation scheme. Let $\mathcal{L}\big(H_0^1(\Omega)\big)$ be a Banach space constituting of a set of bounded linear operators on $H_0^1(\Omega)$ with norm $\|Q\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} := \sup_{0\neq u\in H_0^1(\Omega)} \dfrac{\|Qu\|_{H_0^1(\Omega)}}{\|u\|_{H_0^1(\Omega)}}$ for each $Q \in \mathcal{L}\big(H_0^1(\Omega)\big)$. Therefore, $S_h(\Omega)$ is considered as a finite-dimensional subspace of $H_0^1(\Omega)$ depending on the discretization parameter $h > 0$ with the same inner product and norm as $H_0^1(\Omega)$.

Assumption 1. *Operator $I - A$ is invertible. Namely, there exists $(I - A)^{-1} \in \mathcal{L}\big(H_0^1(\Omega)\big)$.*

Let $P_h \in \mathcal{L}\big(H_0^1(\Omega), S_h(\Omega)\big)$ be an orthogonal projection defined in (4). Then note that $\|P_h\|_{\mathcal{L}\left(H_0^1(\Omega),S_h(\Omega)\right)} \leq 1$ holds. We now assume the following two convergence properties:

Assumption 2. *For an arbitrary $u \in H_0^1(\Omega)$, $P_h u$ converges to u in $H_0^1(\Omega)$ as $h \to 0$.*

Assumption 3. *For each h, there exists a positive constant $C(h)$, which converges to 0 as $h \to 0$, satisfying*

$$\|(I - P_h)Au\|_{H_0^1(\Omega)} \leq C(h)\,\|u\|_{H_0^1(\Omega)}, \quad \forall u \in H_0^1(\Omega).$$

Assumptions 2 and 3 correspond to (5) and (7), respectively, in the previous section. Therefore, as mentioned in subsection 1.2, these assumptions are quite reasonable conditions for usual finite element subspace $S_h(\Omega) \subset H_0^1(\Omega)$.

Remark 1. From the assumptions 1 and 3, there exists a constant $\delta_A > 0$ such that, for all $h \in (0, \delta_A)$,

$$C(h) < \frac{1}{\|(I - A)^{-1}\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}}. \tag{10}$$

Due to the compactness of operator $P_h A \in \mathcal{L}\big(H_0^1(\Omega), S_h(\Omega)\big)$, we have the following properties.

Lemma 1. *Let δ_A be the same constant in Remark 1. Then, for all $h \in (0, \delta_A)$, there exists a bounded inverse of $I - P_h A$ with estimates*

$$\left\|(I - P_h A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} \leq \frac{\left\|(I - A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}}{1 - C(h)\,\|(I - A)^{-1}\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}}. \tag{11}$$

Proof. For an arbitrary $f \in H_0^1(\Omega)$, we consider the solution $u \in H_0^1(\Omega)$ satisfying:

$$(I - P_h A)u = f. \tag{12}$$

From assumption 1, it is readily seen that (12) is equivalent to the following fixed point equation:

$$u = -(I - A)^{-1}(I - P_h)Au + (I - A)^{-1}f =: T_{h,f}(u). \tag{13}$$

Hence, by using assumption 3, for arbitrary v, $w \in H_0^1(\Omega)$, we have

$$\begin{aligned} \|T_{h,f}(v) - T_{h,f}(w)\|_{H_0^1(\Omega)} &= \left\|(I - A)^{-1}(I - P_h)A(v - w)\right\|_{H_0^1(\Omega)} \\ &\le \left\|(I - A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} C(h) \|v - w\|_{H_0^1(\Omega)}. \end{aligned}$$

If h is sufficiently small that (10) holds, then $T_{h,f}$ is a contraction map. Therefore, $T_{h,f}$ has a unique fixed point $u \in H_0^1(\Omega)$ satisfying (12) by Banach's fixed point theorem. Furthermore, the arbitrariness of f implies that $I - P_h A$ is a bijection on $H_0^1(\Omega)$ for such an h.
Also, by some simple calculation using (13) with assumption 3, we obtain

$$\left\|(I - P_h A)^{-1} f\right\|_{H_0^1(\Omega)} \le \frac{\left\|(I - A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}}{1 - \left\|(I - A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} C(h)} \|f\|_{H_0^1(\Omega)},$$

which yields the desired estimates (11). □

Note that

$$(I - P_h A)u_h = P_h(I - A)u_h, \quad \forall u_h \in S_h(\Omega)$$

holds. This fact means that $I - P_h A$ is equal to $P_h(I - A)$ on $S_h(\Omega)$, namely, $(I - P_h A)|_{S_h(\Omega)} = P_h(I - A)|_{S_h(\Omega)}$ holds. Therefore, let define $[I - A]_h \in \mathcal{L}(S_h(\Omega))$ by $[I - A]_h := P_h(I - A)|_{S_h(\Omega)}$. The following lemma gives an invertibility condition of $[I - A]_h$, and estimates for the norm of $[I - A]_h^{-1}$.

Lemma 2. *Under the same conditions as in Lemma 1, for all $h \in (0, \delta_A)$, there exists a inverse of $[I - A]_h$ and the following estimate holds*

$$\left\|[I - A]_h^{-1}\right\|_{\mathcal{L}\left(S_h(\Omega)\right)} \le \left\|(I - P_h A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}. \tag{14}$$

Proof. For an $f_h \in S_h(\Omega)$, if $P_h(I - A)f_h = 0$, then

$$\begin{aligned} f_h &= P_h A f_h \\ &= -(I - P_h)Af_h + Af_h. \end{aligned}$$

Hence we have

$$(I - A)f_h = -(I - P_h)Af_h.$$

Namely,

$$f_h = -(I-A)^{-1}(I-P_h)Af_h.$$

Therefore, by assumption 3, we have

$$\begin{aligned}||f_h||_{H_0^1(\Omega)} &\le ||(I-A)^{-1}||_{\mathcal{L}\left(H_0^1(\Omega)\right)}||(I-P_h)Af_h||_{H_0^1(\Omega)}\\ &\le ||(I-A)^{-1}||_{\mathcal{L}\left(H_0^1(\Omega)\right)}C(h)||f_h||_{H_0^1(\Omega)},\end{aligned}$$

which yields $f_h = 0$ from (10). Taking notice that the existence and uniqueness of the solution are equivalent for the finite dimensional linear equation on $S_h(\Omega)$, the invertibility of $[I-A]_h$ follows immediately.
Next, observe that

$$\begin{aligned}\left\|(I-P_hA)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} &= \sup_{0\neq f\in H_0^1(\Omega)}\frac{\|f\|_{H_0^1(\Omega)}}{\|(I-P_hA)f\|_{H_0^1(\Omega)}}\\ &\ge \sup_{0\neq f_h\in S_h(\Omega)}\frac{\|f_h\|_{H_0^1(\Omega)}}{\|(I-P_hA)f_h\|_{H_0^1(\Omega)}}\\ &= \sup_{0\neq f_h\in S_h(\Omega)}\frac{\|f_h\|_{H_0^1(\Omega)}}{\|P_h(I-A)f_h\|_{H_0^1(\Omega)}}\\ &= \left\|[I-A]_h^{-1}\right\|_{\mathcal{L}\left(S_h(\Omega)\right)},\end{aligned}$$

which completes the proof of (14). □

On the convergence of $(I-P_hA)^{-1}$, we have the following lemma:

Lemma 3. *The following convergence property holds:*

$$\lim_{h\to 0}\left\|(I-P_hA)^{-1}-(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} = 0$$

Proof. Let δ_A be the same constant defined above. Therefore, for each $h\in(0,\delta_A)$, $I-P_hA$ is invertible on $H_0^1(\Omega)$ by lemma 1. For an arbitrary $f\in H_0^1(\Omega)$, we set $u := (I-A)^{-1}f \in H_0^1(\Omega)$ and $w(h) := (I-P_hA)^{-1}f \in H_0^1(\Omega)$. Then we have

$$(I-A)u = f \quad \text{and} \quad (I-P_hA)w(h) = f.$$

Hence, we obtain

$$(I-A)\big(u-w(h)\big) = (I-P_hA)w(h)-(I-A)w(h),$$

which is rewritten as

$$u-w(h) = (I-A)^{-1}(I-P_h)Aw(h).$$

From assumption 3, we obtain

$$\begin{aligned}\|u - w(h)\|_{H_0^1(\Omega)} &\le \left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} C(h) \|w(h)\|_{H_0^1(\Omega)} \\ &\le \left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} C(h) \left(\|u - w(h)\|_{H_0^1(\Omega)} + \|u\|_{H_0^1(\Omega)}\right).\end{aligned}$$

Hence we have

$$\begin{aligned}\left(1 - C(h)\left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}\right)\|u - w(h)\|_{H_0^1(\Omega)} \le \\ C(h)\left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} \|u\|_{H_0^1(\Omega)}.\end{aligned}$$

Taking notice of (10),

$$\|u - w(h)\|_{H_0^1(\Omega)} \le \frac{C(h)\left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}}{1 - C(h)\left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}} \|u\|_{H_0^1(\Omega)}.$$

Namely, it holds that

$$\begin{aligned}&\left\|(I-A)^{-1}f - (I-P_hA)^{-1}f\right\|_{H_0^1(\Omega)} \\ &\le \frac{C(h)\left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}}{1 - C(h)\left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}} \left\|(I-A)^{-1}f\right\|_{H_0^1(\Omega)} \\ &\le \frac{C(h)\left\|(I-A)^{-1}\right\|^2_{\mathcal{L}\left(H_0^1(\Omega)\right)}}{1 - C(h)\left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}} \|f\|_{H_0^1(\Omega)}.\end{aligned}$$

Therefore, we obtain the following convergence property:

$$\left\|(I-P_hA)^{-1} - (I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} \le \frac{C(h)\left\|(I-A)^{-1}\right\|^2_{\mathcal{L}\left(H_0^1(\Omega)\right)}}{1 - C(h)\left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}} \to 0$$

as $h \to 0$, which yields the desired conclusion. □

Theorem 2. *The following convergence property holds for each $f \in H_0^1(\Omega)$.*

$$\lim_{h\to 0}\left\|[I-A]_h^{-1}P_hf - (I-A)^{-1}f\right\|_{H_0^1(\Omega)} = 0. \tag{15}$$

Proof. Let δ_A be a positive constant satisfying condition (10) and let h be a fixed parameter in $(0, \delta_A)$. Then, there exists $[I-A]_h^{-1} \in \mathcal{L}\left(S_h(\Omega)\right)$ by lemma 2. For each $f \in H_0^1(\Omega)$, we set $u := (I-A)^{-1}f \in H_0^1(\Omega)$ and $u_h := [I-A]_h^{-1}P_hf \in S_h(\Omega)$. By the definition, we have

$$\begin{aligned}f - P_hf &= (I-A)u - P_h(I-A)u_h \\ &= (I-P_hA)(u-u_h) + (I-A)u - (I-P_hA)u \\ &= (I-P_hA)(u-u_h) - (I-P_h)Au.\end{aligned}$$

Noting that there also exists $(I - P_h A)^{-1} \in \mathcal{L}\big(H_0^1(\Omega)\big)$ by lemma 1, from the assumption 3 and (11), we obtain, by using the above equality,

$$\begin{aligned}
\|u - u_h\|_{H_0^1(\Omega)} &= \big\|(I - P_h A)^{-1}\big(f - P_h f + (I - P_h)Au\big)\big\|_{H_0^1(\Omega)} \\
&\leq \frac{\big\|(I - A)^{-1}\big\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}}{1 - C(h)\,\big\|(I - A)^{-1}\big\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}} \Big(\|f - P_h f\|_{H_0^1(\Omega)} + C(h)\,\|u\|_{H_0^1(\Omega)}\Big) \\
&\leq \frac{\big\|(I - A)^{-1}\big\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}}{1 - C(h)\,\big\|(I - A)^{-1}\big\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}} \Big(\|f - P_h f\|_{H_0^1(\Omega)} \\
&\qquad + C(h)\,\big\|(I - A)^{-1}\big\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}\,\|f\|_{H_0^1(\Omega)}\Big). \qquad (16)
\end{aligned}$$

The right-hand side of (16) converges to 0 as $h \to 0$ by the assumptions 2 and 3. Thus, (15) is proved. □

Now we present the norm convergence theorem, which is the main result of this paper.

Theorem 3. *The following norm convergence property holds:*

$$\lim_{h \to 0} \big\|[I - A]_h^{-1}\big\|_{\mathcal{L}\left(S_h(\Omega)\right)} = \big\|(I - A)^{-1}\big\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}.$$

Proof. First, note that, for each fixed $f \in H_0^1(\Omega)$, we have by Theorem 2

$$\big\|(I - A)^{-1} f\big\|_{H_0^1(\Omega)} = \lim_{h \to 0} \big\|[I - A]_h^{-1} P_h f\big\|_{S_h(\Omega)}.$$

Therefore, it holds that

$$\begin{aligned}
\big\|(I - A)^{-1}\big\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} &= \sup_{||f||_{H_0^1(\Omega)} = 1} \big\|(I - A)^{-1} f\big\|_{H_0^1(\Omega)} \\
&= \sup_{||f||_{H_0^1(\Omega)} = 1} \lim_{h \to 0} \big\|[I - A]_h^{-1} P_h f\big\|_{S_h(\Omega)}. \qquad (17)
\end{aligned}$$

Moreover, for each $h \in (0, \delta_A)$ and $f \in H_0^1(\Omega)$ with $\|f\|_{H_0^1(\Omega)} = 1$, observe that by using Lemma 2

$$\begin{aligned}
\big\|[I - A]_h^{-1} P_h f\big\|_{S_h(\Omega)} &\leq \big\|[I - A]_h^{-1}\big\|_{\mathcal{L}\left(S_h(\Omega)\right)}\,||P_h f||_{S_h(\Omega)} \\
&\leq \big\|[I - A]_h^{-1}\big\|_{\mathcal{L}\left(S_h(\Omega)\right)} \qquad (18) \\
&\leq ||(I - P_h A)^{-1}||_{\mathcal{L}\left(H_0^1(\Omega)\right)}. \qquad (19)
\end{aligned}$$

On the other hand, by Lemma 3, it holds that the right-hand side of (19) converges to $\big\|(I - A)^{-1}\big\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}$ as $h \to 0$. Combining this fact with (17)-(19) we can show

that $\lim_{h\to 0} \left\|[I-A]_h^{-1}\right\|_{\mathcal{L}\left(S_h(\Omega)\right)}$ exists and equals $\left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}$. Indeed, we take the limit inferior and limit superior of (18) and (19),

$$\begin{aligned}\lim_{h\to 0} \left\|[I-A]_h^{-1}P_h f\right\|_{S_h(\Omega)} &\le \liminf_{h\to 0} \left\|[I-A]_h^{-1}\right\|_{\mathcal{L}\left(S_h(\Omega)\right)} \\ &\le \limsup_{h\to 0} \left\|[I-A]_h^{-1}\right\|_{\mathcal{L}\left(S_h(\Omega)\right)} \\ &\le \left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} \end{aligned} \tag{20}$$

holds. Here, the last inequality follows from Lemma 3. Taking notice that the inequalities, except for the first left-hand sides in (20) is independent of f, we obtain from (17)

$$\begin{aligned}\left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} &= \sup_{||f||_{H_0^1(\Omega)}=1} \lim_{h\to 0} \left\|[I-A]_h^{-1}P_h f\right\|_{S_h(\Omega)} \\ &\le \liminf_{h\to 0} \left\|[I-A]_h^{-1}\right\|_{\mathcal{L}\left(S_h(\Omega)\right)} \\ &\le \left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)}\end{aligned}$$

Combining the above with (20), we have

$$\liminf_{h\to 0} \left\|[I-A]_h^{-1}\right\|_{\mathcal{L}\left(S_h(\Omega)\right)} = \limsup_{h\to 0} \left\|[I-A]_h^{-1}\right\|_{\mathcal{L}\left(S_h(\Omega)\right)} = \left\|(I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)},$$

which yields the desired conclusion. □

Remark 2. Note that the result of Theorem 3 does not mean $[I-A]_h^{-1}P_h \to (I-A)^{-1}$ as $h\to 0$ in $\mathcal{L}\left(H_0^1(\Omega)\right)$.
Actually, if $\lim_{h\to 0} \left\|[I-A]_h^{-1}P_h - (I-A)^{-1}\right\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} = 0$ holds, then considering the particular case: $A \equiv 0$, it implies that $\lim_{h\to 0} \|P_h - I\|_{\mathcal{L}\left(H_0^1(\Omega)\right)} = 0$. From the fact that P_h is a finite dimensional operator, this contradicts that the identity operator I is not compact on the infinite dimensional space $\mathcal{L}\left(H_0^1(\Omega)\right)$.

3 Conclusion

We presented the convergence theorem of $[I-A]_h^{-1}P_h$ to $(I-A)^{-1}$ as $h\to 0$ in Theorem 2, and we also established the norm convergence theorem in Theorem 3. Moreover, Lemma 2 is important as a theoretical result for the existence of the Galerkin approximation for $(I-A)^{-1}$. It is also expected that these results can be extended for the more general linear compact operator A, e.g., corresponding to the biharmonic problems, under similar assumptions to 1, 2, and 3.

Acknowledgment

The authors heartily thank the anonymous referee for her/his thorough reading and valuable comments.

References

[1] Brenner, Susanne C. and Scott, L. Ridgway. *The Mathematical Theory of Finite Element Methods*. Springer, New York, second edition, 2002.

[2] Ciarlet, P.G. and Lions, J.L., editors. *Handbook of Numerical Analysis Volume II, Finite Element Methods (Part 1)*. Elsevier Science B.V., 1990.

[3] Kinoshita, Takehiko, Watanabe, Yoshitaka, and Nakao, Mitsuhiro T. Some remarks on the rigorous estimation of inverse linear elliptic operators. In *International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics*, Volume 9553 of *Lecture Notes in Computer Science*, pages 225–235. Springer, 2016. DOI: 10.1007/978-3-319-31769-4_18.

[4] Kinoshita, Takehiko, Watanabe, Yoshitaka, and Nakao, Mitsuhiro T. Some lower bound estimates for resolvents of a compact operator on an infinite-dimensional Hilbert space. *Journal of Computational and Applied Mathematics*, 369, 2020. DOI: 10.1016/j.cam.2019.112561, 112561.

[5] Nakao, Mitsuhiro T., Hashimoto, Kouji, and Watanabe, Yoshitaka. A numerical method to verify the invertibility of linear elliptic operators with applications to nonlinear problems. *Computing*, 75(1):1–14, 2005. DOI: 10.1007/s00607-004-0111-1.

[6] Nakao, Mitsuhiro T., Plum, Michael, and Watanabe, Yoshitaka. *Numerical Verification Methods and Computer-Assisted Proofs for Partial Differential Equations*. Springer, Singapore, 2019. DOI: 10.1007/978-981-13-7669-6.

[7] Nakao, Mitsuhiro T., Watanabe, Yoshitaka, Kinoshita, Takehiko, Kimura, Takuma, and Yamamoto, Nobito. Some considerations of the invertibility verifications for linear elliptic operators. *Japan Journal of Industrial and Applied Mathematics*, 32:19–31, 2015. DOI: 10.1007/s13160-014-0160-6.

[8] Oden, John T. and Reddy, Junuthula N. *An Introduction to the Mathematical Theory of Finite Elements*. John Wiley & Sons, New York, 1976.

[9] Oishi, Shin'ichi. Numerical verification of existence and inclusion of solutions for nonlinear operator equations. *Journal of Computational and Applied Mathematics*, 60:171–185, 1995. DOI: 10.1016/0377-0427(94)00090-N.

[10] Plum, Michael. Eigenvalue inclusions for second-order ordinary differential operators by a numerical homotopy method. *Zeitschrift für angewandte Mathematik und Physik (ZAMP)*, 41:205–226, 1990. DOI: 10.1007/BF00945108.

[11] Rump, Siegfried M. INTLAB — INTerval LABoratory. In Csendes, Tibor, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. URL: http://www.ti3.tuhh.de/rump/.

[12] Watanabe, Yoshitaka, Kinoshita, Takehiko, and Nakao, Mitsuhiro T. A posteriori estimates of inverse operators for boundary value problems in linear elliptic partial differential equations. *Mathematics of Computation*, 82:1543–1557, 2013. DOI: 10.1090/S0025-5718-2013-02676-2.

Acta Cybernetica **26** (2023) 83–103.

B_π^R-Matrices, B-Matrices, and Doubly B-Matrices in the Interval Setting*

Matyáš Lorenc[a]

Abstract

In this paper, we focus on generalizing B_π^R-matrices into the interval setting, including some results regarding this class. There are two possible ways to generalize B_π^R-matrices into the interval setting, but we prove that, in a sense, they are one. We derive mainly recognition methods for this interval matrix class, such as characterizations, necessary conditions, and sufficient ones.

Next, we also take a look at interval B-matrices and interval doubly B-matrices, which were introduced recently, and we present characterizations through reduction for them and for B_π^R-matrices.

Keywords: B_π^R-matrix, B-matrix, doubly B-matrix, interval analysis, interval matrix, P-matrix

1 Introduction

***P*-matrices.** An important class of matrices, in optimization as well as linear algebra and graph theory (see [7]), is the class of P-matrices. Recall that $A \in \mathbb{R}^{n\times n}$ is a *P-matrix* if all its principal minors (i.e. determinants of its principal submatrices) are positive.

The class of P-matrices has a close connection to the linear complementarity problem (which is more thoroughly described in [1]), which is one of the reasons the P-matrices are studied. A connection has even been found between P-matrices and the regularity of interval matrices, as shown in [5] or [16]. However, the task of verifying whether a given matrix is a P-matrix is co-NP-complete, as proved in [2].

***B*-matrices, Doubly *B*-matrices, B_π^R-matrices.** Testing P-matrix property is hard; it is important to identify such subclasses of P-matrices which are efficiently

*Supported by the Czech Science Foundation Grant P403-22-11117S.

[a]Charles University, Faculty of Mathematics and Physics, Department of Applied Mathematics, Malostranské nám. 25, 11800, Prague, Czech Republic, E-mail: lorenc@kam.mff.cuni.cz, ORCID: 0000-0002-6797-9052

DOI: 10.14232/actacyb.294423

recognizable. Besides positive definite matrices or M-matrices, those might be e.g., B-matrices (introduced in [14]), doubly B-matrices (introduced in [15]) or B_π^R-matrices (introduced in [12]); here we will focus mainly on the last mentioned. In addition to their usefulness as subclasses of the P-matrices, these matrix classes also appeared in the context of Markov chains and in localization of eigenvalues.

Interval analysis. Interval analysis was developed to deal with inaccuracy in data, rounding errors, or a certain form of uncertainty. A central concept of interval analysis is an interval matrix. We denote the set of all real intervals by $\mathbb{IR}$. Now, let us define an interval matrix.

Definition 1.1 (Interval matrix)**.** *An* interval matrix $\boldsymbol{A}$*, which we denote by* $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$*, is defined as*

$$\boldsymbol{A} = \left[\underline{A}, \overline{A}\right] = \left\{ A \in \mathbb{R}^{m\times n} \,\middle|\, \underline{A} \leq A \leq \overline{A} \right\},$$

where $\underline{A}, \overline{A}$ *are called the lower or upper bound matrices of* $\boldsymbol{A}$*, respectively, and* $\leq$ *is understood entrywise.*

We can look at $\boldsymbol{A}$ *as a matrix with its entries from* $\mathbb{IR}$*, hence* $\forall i \in [m], \forall j \in [n] : \boldsymbol{a}_{ij} = \left[\underline{a}_{ij}, \overline{a}_{ij}\right]$*, where* $[m] = \{1, 2, \ldots, m\}$ *and analogously for* $[n]$*.*

Definition 1.2. *Let* $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$*. We say that* $\boldsymbol{A}$ *has positive row sums if the intervals of the row sums are positive. In other words, if* $\forall i \in [m] : \sum_{j=1}^{n} \underline{a}_{ij} > 0$*.*

We call an interval matrix $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ an *interval P-matrix* if every $A \in \boldsymbol{A}$ is a P-matrix. Similarly other matrix classes might be defined, e.g., the class of Z-matrices, which are matrices with non-positive off-diagonal elements. We can also define some basic properties, such as regularity, which are studied more in the following works: [3], [6], [8], and [9], among many others.

Structure and contribution of the paper. In this work, we present some results based on [10], such as a generalization of B_π^R-matrices into the interval settings, and lay the foundations for recognizing the interval variants through characterization, or sufficient conditions and necessary ones. We then proceed to introduce characterizations through reduction of interval B-matrices, doubly B-matrices, and B_π^R-matrices.

As we show, these interval variants of our matrix classes are connected to the interval P-matrices in the same way the real variants are connected to the real P-matrices. Interval P-matrices are closely connected to the linear complementarity problem with uncertain data, which might be modeled by intervals. So again, it is useful to have easily recognizable subclasses of interval P-matrices.

2 B^R_π-matrices

2.1 Real B^R_π-matrices

Let us start by introducing real B^R_π-matrices and a few facts about them, which were introduced by Neumann, Peña, and Pryporova in [12] or by Araújo and Mendes-Gonçalves in [11], and which we will later transfer into the interval setting.

Definition 2.1 (B^R_π-matrix, [12])**.** *Let $A \in \mathbb{R}^{n\times n}$, let $\pi \in \mathbb{R}^n$ such that it fulfills*

$$0 < \sum_{j=1}^{n} \pi_j \leq 1, \tag{1}$$

*and let $R \in \mathbb{R}^n$ be the vector formed by the row sums of A (hence $\forall i \in [n] : \ R_i = \sum_{j=1}^{n} a_{ij}$). We say that A is a B^R_π-*matrix *if $\forall i \in [n]$:*

$$\begin{aligned} &a) \quad R_i > 0 \\ &b) \quad \forall k \in [n] \setminus \{i\} : \quad \pi_k \cdot R_i > a_{ik} \end{aligned}$$

The next proposition is introduced in [12] as Observation 3.2.

Proposition 2.1. *Let $A \in \mathbb{R}^{n\times n}$ have positive row sums, and let $R \in \mathbb{R}^n$ be the vector formed by the row sums of A. There exists a vector $\pi \in \mathbb{R}^n$ satisfying inequality* (1) *such that A is a B^R_π-matrix if and only if*

$$\sum_{j=1}^{n} \max\left\{ \frac{a_{ij}}{R_i} \middle| \, i \neq j \right\} < 1.$$

Remark 2.1. If for any matrix $A \in \mathbb{R}^{n\times n}$ the condition from Proposition 2.1 is satisfied, then we are able to construct a vector $\pi \in \mathbb{R}^n$ satisfying inequality (1) such that A is a B^R_π-matrix in the following manner:

1. We define $\epsilon \in \mathbb{R}$ as
$$\epsilon = 1 - \sum_{j=1}^{n} \max\left\{ \frac{a_{ij}}{R_i} \middle| \, i \neq j \right\},$$
2. and then for every $j \in [n]$ we define π_j as
$$\pi_j = \max\left\{ \frac{a_{ij}}{R_i} \middle| \, i \neq j \right\} + \frac{\epsilon}{n}.$$

Of course, instead of $\frac{\epsilon}{n}$ in the second step we may use any constant $0 < c \leq \frac{\epsilon}{n}$, or we might use a vector $\xi \in \mathbb{R}^{+^n}$ such that $0 < \sum_{j=1}^{n} \xi_j \leq \epsilon$, and define π_j as

$$\pi_j = \max\left\{ \frac{a_{ij}}{R_i} \middle| \, i \neq j \right\} + \xi_j.$$

(It is easy to verify that this holds from Definition 2.1, because so defined π meets condition b) for the above-mentioned definition, and also satisfies inequality (1).)

The following result is stated and proved in [13].

Proposition 2.2. *Every B_π^R-matrix with $\pi \geq 0$ is a P-matrix.*

Remark 2.2. We can show an example of a B_π^R-matrix with $\pi_i < 0$ for some $i \in [n]$, which is not a B_ψ^R-matrix for any $\psi \geq 0$. (To verify this fact, the reader may use the properties of B_π^R-matrices stated in the next proposition, more precisely, part 1).)

Example 2.1.

$$A = \begin{pmatrix} \frac{3}{2} & -1 \\ 2 & -\frac{1}{2} \end{pmatrix}$$

It is easy to check that A is a B_π^R-matrix for $\pi = (2, -1)^T$. (And it is clearly not a P-matrix.)

Hence, for the purpose of this work, we are interested only in such B_π^R-matrices that have $\pi \geq 0$, since only those ought to be P-matrices.

The next proposition is introduced in [11] as Proposition 2.1.

Proposition 2.3. *Let $\pi \in \mathbb{R}^n$ such that inequality (1) holds, and let $A \in \mathbb{R}^{n\times n}$ be a B_π^R-matrix, where $R \in \mathbb{R}^n$ is the vector of row sums of A. Then the following holds:*

1. $\forall i \in [n]: \quad a_{ii} > \pi_i \cdot R_i,$
2. $\forall (i,j) \in [n]^2, j \neq i: \quad \pi_i \geq \pi_j \;\Rightarrow\; a_{ii} > a_{ij},$
3. *let* $k = \operatorname{argmax}\{\pi_i \mid i \in [n]\}$, *then* $\forall j \neq k: \quad a_{kk} > a_{kj}$, *and*
4. $\forall (i,j) \in [n]^2, j \neq i: \quad \pi_j \leq 0 \;\Rightarrow\; a_{ij} < 0.$

The next proposition is introduced in [11] as Proposition 2.5.

Proposition 2.4. *Let $\pi \in \mathbb{R}^n$ such that condition (1) holds, and let $A \in \mathbb{R}^{n\times n}$ be a B_π^R-matrix. If $\alpha \in \mathbb{R}^n$ satisfies analogy of inequality (1) and $\alpha \geq \pi$, then A is a B_α^R-matrix.*

2.2 Interval $B_\pi^{\boldsymbol{R}}$-matrices

Next, we proceed to generalize the class of B_π^R-matrices into the interval setting. However, there are two ways to do so, differing in the order of quantifiers.

Definition 2.2 (Homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrix)**.** *Let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$, $\pi \in \mathbb{R}^n$ such that inequality (1) holds, and let $\boldsymbol{R} \in \mathbb{IR}^n$. We say that $\boldsymbol{A}$ is a* homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrix *if $\forall A \in \boldsymbol{A}$: $\exists R \in \boldsymbol{R}$ such that A is a (real) B_π^R-matrix.*

Here, the $\boldsymbol{R}$ in the definition can be perceived as the vector whose entries correspond to the intervals of the row sums of matrices $A \in \boldsymbol{A}$, but in the interval setting it is more of a symbol than of any greater significance. This is because if we

have two interval $B_\pi^{\boldsymbol{R}}$-matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, we cannot say that any two $A \in \boldsymbol{A}$ and $B \in \boldsymbol{B}$ are real B_π^R-matrices for the same R. Despite that, we decided to include it in the notation of the interval matrix class for compatibility with the real case definition.

Corollary 2.1. *Every homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrix with $\pi \geq 0$ is an interval P-matrix.*

Proof. It holds for every instance, hence it holds for the whole interval matrix. □

Definition 2.3 (Heterogeneous interval $B_\Pi^{\boldsymbol{R}}$-matrix)**.** *Let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$, and let $\boldsymbol{R} \in \mathbb{IR}^n$. We say that $\boldsymbol{A}$ is a* heterogeneous interval $B_\Pi^{\boldsymbol{R}}$-matrix *if $\forall A \in \boldsymbol{A}$: $\exists R \in \boldsymbol{R}, \exists \pi \in \mathbb{R}^n$ such that condition* (1) *holds and A is a (real) B_π^R-matrix.*

Here, the $\boldsymbol{R}$ in the definition again has the same meaning as in the case of homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrices. As for the Π, we may understand it as a set of all such vectors π satisfying condition (1) such that there exists $A \in \boldsymbol{A}$, for which it holds that A is a real B_π^R-matrix. However, again it can be perceived just as a symbol that distinguishes this interval matrix class, since the exact form or content of the set Π holds no real significance to us, and we have no way of deriving it yet.

Corollary 2.2. *Every homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrix is a heterogeneous interval $B_\Pi^{\boldsymbol{R}}$-matrix.*

Proof. It trivially follows from the definitions. □

Let us start by stating a characterization that helps us with the recognition of the class of homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrices in finite time.

Theorem 2.1. *Let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$, let $\pi \in \mathbb{R}^n$ satisfy inequality* (1)*, and let $\boldsymbol{R} \in \mathbb{IR}^n$ be the vector of intervals of individual row sums in matrix $\boldsymbol{A}$. The matrix $\boldsymbol{A}$ is a homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrix if and only if $\forall i \in [n]$ the following properties hold:*

a) $\underline{R}_i > 0$

b) $\forall k \in [n] \setminus \{i\}$:

$$\left(\pi_k > 1 \quad \Rightarrow \quad \sum_{j\neq k} \underline{a}_{ij} \;>\; \left(\frac{1}{\pi_k} - 1\right) \cdot \underline{a}_{ik}\right)$$

$$\wedge \quad \left(0 < \pi_k \leq 1 \quad \Rightarrow \quad \sum_{j\neq k} \underline{a}_{ij} \;>\; \left(\frac{1}{\pi_k} - 1\right) \cdot \overline{a}_{ik}\right)$$

$$\wedge \quad \left(\pi_k = 0 \quad \Rightarrow \quad 0 \;>\; \overline{a}_{ik}\right)$$

$$\wedge \quad \left(\pi_k < 0 \quad \Rightarrow \quad \sum_{j\neq k} \overline{a}_{ij} \;<\; \left(\frac{1}{\pi_k} - 1\right) \cdot \overline{a}_{ik}\right)$$

Proof. Condition a) of Definition 2.1 evaluated for every $A \in \boldsymbol{A}$ is equivalent to $\underline{R}_i > 0$. As for the condition b) of the definition, it may be modified for every $k \neq i$ as follows (while noting that $\pi_k \cdot R_i = \pi_k \cdot \sum_{j=1}^{n} a_{ij}$):

1. $\pi_k > 1$:

$$\pi_k \cdot \sum_{j=1}^{n} a_{ij} > a_{ik} \quad \Leftrightarrow \quad \sum_{j=1}^{n} a_{ij} > \frac{1}{\pi_k} \cdot a_{ik} \quad \Leftrightarrow \quad \sum_{j \neq k} a_{ij} > \left(\frac{1}{\pi_k} - 1\right) \cdot a_{ik} \tag{2}$$

We observe that the highest value of $\left(\frac{1}{\pi_k} - 1\right) \cdot a_{ik}$ is attained at the lower bound on the $\boldsymbol{a}_{ik}$, because when $\pi_k > 1$, we have $\left(\frac{1}{\pi_k} - 1\right) < 0$. Whence, the condition above holds for every $A \in \boldsymbol{A}$ if and only if the following condition holds:

$$\sum_{j \neq k} \underline{a}_{ij} > \left(\frac{1}{\pi_k} - 1\right) \cdot \underline{a}_{ik}$$

2. $0 < \pi_k \leq 1$: Using the chain of equivalences (2) from the previous part, we observe that the highest value of $\left(\frac{1}{\pi_k} - 1\right) \cdot a_{ik}$ is obtained by the upper bound on the $\boldsymbol{a}_{ik}$, because when $0 < \pi_k \leq 1$, then $\left(\frac{1}{\pi_k} - 1\right) \geq 0$. Thus, condition b) of Definition 2.1 holds for every $A \in \boldsymbol{A}$ if and only if the following condition holds:

$$\sum_{j \neq k} \underline{a}_{ij} > \left(\frac{1}{\pi_k} - 1\right) \cdot \overline{a}_{ik}$$

3. $\pi_k = 0 : \pi_k \cdot \sum_{j=1}^{n} a_{ij} > a_{ik} \quad \Leftrightarrow \quad 0 > a_{ik}$
The condition above holds for every $A \in \boldsymbol{A}$ if and only if $0 > \overline{a}_{ik}$

4. $\pi_k < 0$:

$$\pi_k \cdot \sum_{j=1}^{n} a_{ij} > a_{ik} \quad \Leftrightarrow \quad \sum_{j=1}^{n} a_{ij} < \frac{1}{\pi_k} \cdot a_{ik} \quad \Leftrightarrow \quad \sum_{j \neq k} a_{ij} < \left(\frac{1}{\pi_k} - 1\right) \cdot a_{ik}$$

We observe that the smallest value of $\left(\frac{1}{\pi_k} - 1\right) \cdot a_{ik}$ is obtained by the upper bound on the $\boldsymbol{a}_{ik}$, because when $\pi_k < 0$, then $\left(\frac{1}{\pi_k} - 1\right) < 0$. From that we have that the condition above holds for every $A \in \boldsymbol{A}$ if and only if the following condition holds:

$$\sum_{j \neq k} \overline{a}_{ij} < \left(\frac{1}{\pi_k} - 1\right) \cdot \overline{a}_{ik} \qquad \square$$

Remark 2.3. This characterization has time complexity $O(n^2)$, which is, surprisingly, the same as a characterization from the definition of the real case, Definition

2.1 (although the interval case has undoubtedly higher implementational complexity).

Let us now introduce an analogy of Proposition 2.1 for homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrices.

Theorem 2.2. *If $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ has positive row sums, then there exists a vector $\pi \in \mathbb{R}^n$ satisfying inequality* (1) *such that $\boldsymbol{A}$ is a homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrix if and only if*

$$\sum_{j=1}^{n} \max \left\{ \frac{\overline{a}_{ij}}{\overline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}}, \frac{\underline{a}_{ij}}{\underline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}} \middle| i \neq j \right\} < 1. \tag{3}$$

Proof. "⇒": $\boldsymbol{A}$ is a $B_\pi^{\boldsymbol{R}}$-matrix for some π satisfying the property (1), hence every $A \in \boldsymbol{A}$ is a B_π^R-matrix, thus, in particular, matrices $A_j \in \boldsymbol{A}$ for every $j \in [n]$ defined as follows:

$$A_j = (a'_{m_1m_2});$$
$$a'_{m_1m_2} = \begin{cases} \overline{a}_{m_1m_2} & \text{if } m_2 = j \ \wedge \ \frac{\overline{a}_{m_1j}}{\overline{a}_{m_1j} + \sum\limits_{m\neq j} \underline{a}_{m_1m}} > \frac{\underline{a}_{m_1j}}{\underline{a}_{m_1j} + \sum\limits_{m\neq j} \underline{a}_{m_1m}}, \\ \underline{a}_{m_1m_2} & \text{otherwise.} \end{cases} \tag{4}$$

Therefore, (if we denote R^j the vector of row sums of A_j) we have

$$\forall j \in [n]:$$
$$\max \left\{ \frac{\overline{a}_{ij}}{\overline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}}, \frac{\underline{a}_{ij}}{\underline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}} \middle| i \neq j \right\} = \max \left\{ \frac{a'_{ij}}{R_i^j} \middle| i \neq j \right\} < \pi_j. \tag{5}$$

But then

$$\sum_{j=1}^{n} \max \left\{ \frac{\overline{a}_{ij}}{\overline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}}, \frac{\underline{a}_{ij}}{\underline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}} \middle| i \neq j \right\} < \sum_{j=1}^{n} \pi_j \leq 1. \tag{6}$$

"⇐": Let

$$\epsilon = 1 - \sum_{j=1}^{n} \max \left\{ \frac{\overline{a}_{ij}}{\overline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}}, \frac{\underline{a}_{ij}}{\underline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}} \middle| i \neq j \right\} > 0,$$

and for every $j \in [n]$ set the $\pi_j = \max \left\{ \frac{\overline{a}_{ij}}{\overline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}}, \frac{\underline{a}_{ij}}{\underline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}} \middle| i \neq j \right\} + \frac{\epsilon}{n}$. Then

$\boldsymbol{A}$ is a homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrix. That is because for any $A \in \boldsymbol{A}$

$$\max\left\{ \frac{\overline{a}_{ij}}{\overline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}}, \frac{\underline{a}_{ij}}{\underline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}} \middle| i \neq j \right\}$$
$$\geq \quad \max\left\{ \frac{a_{ij}}{a_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}} \middle| i \neq j \right\} \quad \geq \quad \max\left\{ \frac{a_{ij}}{\sum\limits_{m=1}^{n} a_{im}} \middle| i \neq j \right\}.$$

Thus, for every $A \in \boldsymbol{A}$ and for every $(k, j) \in [n]^2, j \neq k$, it holds that

$$\frac{a_{kj}}{R_k} \quad = \quad \frac{a_{kj}}{\sum\limits_{m=1}^{n} a_{km}} \quad \leq \quad \max\left\{ \frac{a_{ij}}{\sum\limits_{m=1}^{n} a_{im}} \middle| i \neq j \right\}$$
$$\leq \quad \max\left\{ \frac{\overline{a}_{ij}}{\overline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}}, \frac{\underline{a}_{ij}}{\underline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}} \middle| i \neq j \right\}$$
$$< \quad \max\left\{ \frac{\overline{a}_{ij}}{\overline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}}, \frac{\underline{a}_{ij}}{\underline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}} \middle| i \neq j \right\} + \frac{\epsilon}{n} \quad = \quad \pi_j,$$

ergo $\pi_j \cdot R_k > a_{kj}$. Therefore, every $A \in \boldsymbol{A}$ is a B_π^R-matrix. □

Remark 2.4. If any matrix $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ satisfies the condition from Theorem 2.2, we can construct a vector $\pi \in \mathbb{R}^n$ satisfying condition (1) such that $\boldsymbol{A}$ is a homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrix in an analogous way to what we did in Remark 2.1.

Next, let us introduce one interesting fact about the class of heterogeneous interval B_Π^R-matrices that helps us to characterize it. For that, we first need to state a few auxiliary propositions.

Proposition 2.5. *Let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$. The matrix $\boldsymbol{A}$ is a heterogeneous interval $B_\Pi^{\boldsymbol{R}}$-matrix only if*

$$\sum_{j=1}^{n} \max\left\{ \frac{\overline{a}_{ij}}{\overline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}}, \frac{\underline{a}_{ij}}{\underline{a}_{ij} + \sum\limits_{m\neq j} \underline{a}_{im}} \middle| i \neq j \right\} < 1.$$

Proof. $\boldsymbol{A}$ is a heterogeneous $B_\Pi^{\boldsymbol{R}}$-matrix, hence every $A \in \boldsymbol{A}$ is a B_π^R-matrix for some $\pi = (\pi_1, \ldots, \pi_n)$ satisfying the property (1), thus in particular the matrices $A_j \in \boldsymbol{A}$ for every $j \in [n]$ defined as in the proof of Theorem 2.2 in expression (4) are B_π^R-matrices.

Therefore, (if we denote R^j the vector of row sums of A_j) again just as in the proof of Theorem 2.2, the expression (5) holds $\forall j \in [n]$. From that we also get that expression (6) from the proof holds, which is exactly what we wanted to prove here. □

Corollary 2.3. *Every heterogeneous interval $B_\Pi^{\boldsymbol{R}}$-matrix is a homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrix for some π fulfilling inequality* (1).

Proof. Let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ have positive row sums. From Proposition 2.5, we get the following implication:

$\boldsymbol{A}$ is a heterogeneous interval $B_\Pi^{\boldsymbol{R}}$-matrix $\Rightarrow$ Inequality (3) holds.

From the equivalence from Theorem 2.2 we use the following implication:

Inequality (3) holds $\Rightarrow$ $\exists\pi : \pi$ satisfies condition (1) $\wedge$ $\boldsymbol{A}$ is a homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrix.

Ergo we compose these two implications (because from Definition 2.3 we can easily observe that if $\boldsymbol{A}$ is a heterogeneous interval $B_\Pi^{\boldsymbol{R}}$-matrix, then it has positive row sums, therefore fulfilling the assumptions of Theorem 2.2), and thus obtain the desired implication. □

What we obtained is the second inclusion we need to show the equality among our two interval matrix classes, the class of homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrices and that of the heterogeneous interval $B_\Pi^{\boldsymbol{R}}$-matrices.

Theorem 2.3. *Let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ have positive row sums. We have that $\boldsymbol{A}$ is a heterogeneous interval $B_\Pi^{\boldsymbol{R}}$-matrix if and only if $\exists\pi \in \mathbb{R}^n$ such that condition* (1) *holds and that $\boldsymbol{A}$ is a homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrix.*

Proof. Follows from Corollaries 2.2 and 2.3. □

We proved that the two classes we have defined at the beginning of this subsection are the same, hence it does not make any sense to differentiate the two. Thus, from now on we refer to them as *interval $B_\pi^{\boldsymbol{R}}$-matrices.*

Definition 2.4 (Interval $B_\pi^{\boldsymbol{R}}$-matrix)**.** *Let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$, and let $\pi \in \mathbb{R}^n$ satisfy inequality* (1)*. We say that $\boldsymbol{A}$ is an* interval $B_\pi^{\boldsymbol{R}}$-matrix *if it is a homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrix.*

Remark 2.5. Because of this definition, we can use the same characterizations we use to characterize the homogeneous interval $B_\pi^{\boldsymbol{R}}$-matrices (Theorem 2.1, Theorem 2.2) to characterize the interval $B_\pi^{\boldsymbol{R}}$-matrices (and because of Theorem 2.3 also the $B_\Pi^{\boldsymbol{R}}$-matrices).

Now, let us generalize some properties of real B_π^R-matrices to the interval $B_\pi^{\boldsymbol{R}}$-matrices. The first is a direct consequence of the definition.

Corollary 2.4. *Every interval $B_\pi^{\boldsymbol{R}}$-matrix with $\pi \geq 0$ is an interval P-matrix.*

Proposition 2.6. *Let $\pi \in \mathbb{R}^n$ such that inequality* (1) *is fulfilled, and let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ be an interval $B_\pi^{\boldsymbol{R}}$-matrix. The following holds:*

1. $\forall i \in [n]: \quad \underline{a}_{ii} > \max\left\{\pi_i \cdot \left(\underline{a}_{ii} + \sum_{j \neq i} \underline{a}_{ij}\right), \pi_i \cdot \left(\underline{a}_{ii} + \sum_{j \neq i} \overline{a}_{ij}\right)\right\},$
2. $\forall (i,j) \in [n]^2, j \neq i: \quad \pi_i \geq \pi_j \Rightarrow \underline{a}_{ii} > \overline{a}_{ij},$
3. *if* $k = \operatorname{argmax}\{\pi_i \mid i \in [n]\}$, *then* $\forall j \neq k: \quad \underline{a}_{kk} > \overline{a}_{kj}$, *and*
4. $\forall (i,j) \in [n]^2, j \neq i: \quad \pi_j \leq 0 \Rightarrow \overline{a}_{ij} < 0.$

Proof. Let $\boldsymbol{A} \in \mathbb{IR}^{n \times n}$ be an interval $B_\pi^{\boldsymbol{R}}$-matrix for some $\pi \in \mathbb{R}^n$ fulfilling inequality (1).

1. Let $A_1, A_2 \in \mathbb{R}^{n \times n}$ be defined as follows:

$$A_1 = \underline{A}$$

$$A_2 = (a_{m_1 m_2}); \quad a_{m_1 m_2} = \begin{cases} \underline{a}_{ii} & \text{if } m_1 = m_2 = i, \\ \overline{a}_{m_1 m_2} & \text{otherwise.} \end{cases}$$

   Because $A_1, A_2 \in \boldsymbol{A}$, they are both B_π^R-matrices, thus from Proposition 2.3, part 1) we get that this point holds.

2. Let $A' \in \mathbb{R}^{n \times n}$ be defined as $A' = A_2$, where A_2 is defined in the previous part of this proof. Because $A' \in \boldsymbol{A}$, it is a B_π^R-matrix, thus from Proposition 2.3, part 2) we get that this point holds.

3. Direct consequence of the previous point.

4. Because $\overline{A} \in \boldsymbol{A}$, it is a B_π^R-matrix, thus from Proposition 2.3, part 4) we get that this point holds. □

Proposition 2.7. *Let* $\pi \in \mathbb{R}^n$ *fulfill inequality* (1)*, and let* $\boldsymbol{A} \in \mathbb{IR}^{n \times n}$ *be an interval* $B_\pi^{\boldsymbol{R}}$*-matrix. If* $\alpha \in \mathbb{R}^n$ *satisfies the analogy of inequality* (1) *and* $\alpha \geq \pi$*, then* $\boldsymbol{A}$ *is an interval* $B_\alpha^{\boldsymbol{R}}$*-matrix.*

Proof. It holds for every instance of the interval matrix (see Proposition 2.4), thus it holds for the whole interval matrix. □

3 Characterizations through reduction

Here, in this section, we take a closer look at how we may characterize B_π^R-matrices, B-matrices, and doubly B-matrices through reduction. By that we mean testing an interval matrix for the property of being an interval B_π^R-matrix, B-matrix or doubly B-matrix, respectively, using only a finite subset of instances of the interval matrix, and testing them on being a member of the corresponding real matrix class. Reductions for other matrix classes were surveyed, e.g., by Garloff et al. in [4].

Both the class of interval B-matrices and the one of interval doubly B-matrices were introduced in [10], and we use the characterizations stated and proved there in our proofs. However, everything we use is also stated here as well.

3.1 $B^{\boldsymbol{R}}_{\pi}$-matrices

Let us begin with the interval $B^{\boldsymbol{R}}_{\pi}$-matrices we introduced in section 2.

Proposition 3.1. *Let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$, let $\pi \in \mathbb{R}^n$ satisfy inequality (1), and let $\boldsymbol{R} \in \mathbb{IR}^n$ be the vector of intervals of the individual row sums in matrix $\boldsymbol{A}$. Let $\forall i \in [n] : A_i \in \mathbb{R}^{n\times n}$ be defined as follows:*

1. *if $\pi_i > 1$, then:*
   $A_i = \underline{A}$
2. *else if $0 \leq \pi_i \leq 1$, then:*
   $$A_i = (a_{m_1m_2}); \quad a_{m_1m_2} = \begin{cases} \overline{a}_{m_1m_2} & \text{if } m_1 \neq i, m_2 = i, \\ \underline{a}_{m_1m_2} & \text{otherwise.} \end{cases}$$
3. *else if $\pi_i < 0$, then:*
   $$A_i = (a_{m_1m_2}); \quad a_{m_1m_2} = \begin{cases} \underline{a}_{m_1m_2} & \text{if } m_1 = i, \\ \overline{a}_{m_1m_2} & \text{otherwise.} \end{cases}$$

It holds that $\boldsymbol{A}$ is an interval $B^{\boldsymbol{R}}_{\pi}$-matrix if and only if $\forall i \in [n] :$ A_i is a B^R_π-matrix, where $R \in \mathbb{R}^n$ is the vector of values corresponding to the row sums of A_i.

Proof. "$\Rightarrow$" This holds, because $\forall i \in [n] : A_i \in \boldsymbol{A}$ (and the corresponding $R \in \boldsymbol{R}$).
"$\Leftarrow$"
a) $\forall i \in [n] : \underline{R}_i > 0$, because A_i is a B^R_π-matrix, and $(A_i)_{i,*} = (\underline{A})_{i,*}$, the entries of $\underline{R}$ are positive.
b) $\forall i \in [n]\ \forall k \neq i :$ A_k is a $B^{\boldsymbol{R}}_{\pi}$-matrix and so, from Definition 2.1:

1. $\pi_k > 1 :$

$$\pi_k \cdot \sum_{j=1}^{n} (A_k)_{ij} > (A_k)_{ik} \quad \Leftrightarrow \quad \pi_k \cdot \sum_{j=1}^{n} \underline{a}_{ij} > \underline{a}_{ik}$$
$$\Leftrightarrow \quad \sum_{j\neq k} \underline{a}_{ij} > \left(\frac{1}{\pi_k} - 1\right) \cdot \underline{a}_{ik}$$

2. $0 < \pi_k \leq 1 :$ $\quad \pi_k \cdot \sum_{j=1}^{n} (A_k)_{ij} > (A_k)_{ik} \quad \Leftrightarrow \quad \sum_{j\neq k} \underline{a}_{ij} > \left(\frac{1}{\pi_k} - 1\right) \cdot \overline{a}_{ik}$
3. $\pi_k = 0 :$ $\quad \pi_k \cdot \sum_{j=1}^{n} (A_k)_{ij} > (A_k)_{ik} \quad \Leftrightarrow \quad 0 > \overline{a}_{ik}$
4. $\pi_k < 0 :$ $\quad \pi_k \cdot \sum_{j=1}^{n} (A_k)_{ij} > (A_k)_{ik} \quad \Leftrightarrow \quad \sum_{j\neq k} \overline{a}_{ij} < \left(\frac{1}{\pi_k} - 1\right) \cdot \overline{a}_{ik}$

Thus, $\boldsymbol{A}$ fulfills the conditions of Theorem 2.1, and so it is an interval B^R_π-matrix. □

Proposition 3.2. *The characterization of the interval $B^{\boldsymbol{R}}_{\pi}$-matrices through the reduction given by Proposition 3.1 is for $\pi \geq 0$ minimal with respect to inclusion.*

Proof. First, we notice that from the condition (1) on π, it follows that $\forall j \in [n] : 0 \leq \pi_j \leq 1$, so every matrix from the reduction has the form given by point 2).

If we skip any A_i for arbitrary $i \in [n]$, then we could construct a counterexample, e.g., a unit matrix with interval $[0, \frac{\pi_i}{1-\pi_i}]$ at position (j, i) for arbitrary $j \neq i$. Then $\forall k \neq i : A_k = I_n$, which surely is a B_π^R-matrix. But A_i does not fulfill condition b) from Definition 2.1 in the j-th row. That is because the sum of the j-th row is equal to $1 + \frac{\pi_i}{1-\pi_i}$ and $(A_i)_{ji} = \frac{\pi_i}{1-\pi_i}$, so we get

$$\pi_i \cdot R_j = \pi_i \cdot \left(1 + \frac{\pi_i}{1-\pi_i}\right) = \pi_i \cdot \left(\frac{1-\pi_i+\pi_i}{1-\pi_i}\right) = \frac{\pi_i}{1-\pi_i} = (A_i)_{ji}\,,$$

which violates the condition, and so the A_i is not a B_π^R-matrix. □

Remark 3.1. In Proposition 3.2, the assumption that $\pi \geq 0$ is present both because such a π is what we are interested in in this work, and, more importantly, because for the general case we might have such a π that two entries of the vector are larger than 1. However, then the two matrices A_i corresponding to those entries are the same and equal to $\underline{A}$, and so we may remove one of the two matrices from the reduction, and it still works. As for the case of almost general π, where we only want that there is at most one entry larger than one, we have not managed to prove or disprove the statement yet.

Remark 3.2. This reduction reduces the problem of verifying whether any given interval matrix is an interval $B_\pi^{\boldsymbol{R}}$-matrix, into testing whether n matrices are real B_π^R-matrices.

Example 3.1. Here we show an example of an interval B_π^R-matrix, and use it to point out some things. Let us have a vector π, such that $\pi = (0.36, 0.28, 0.36)$, and let us define an interval B_π^R-matrix $\boldsymbol{A} \in \mathbb{IR}^{3\times 3}$ as follows:

$$\boldsymbol{A} = \begin{pmatrix} [7.95, 8.05] & [-7.05, -6.95] & [-0.05, 0.05] \\ [0.95, 1.05] & [0.95, 1.05] & [0.95, 1.05] \\ [8.95, 9.05] & [10.95, 11.05] & [19.95, 20.05] \end{pmatrix}$$

It is easy to verify that $\boldsymbol{A}$ belongs to the class of B_π^R-matrices for some π satisfying the condition (1) by using Theorem 2.2 or to verify whether it is a B_π^R-matrix for our value of π using Theorem 2.1.

What is quite interesting and important is the fact that this matrix is not positive definite (it is not symmetric), it is not an interval M-matrix (it is not a Z-matrix), nor is it an interval H-matrix (e.g., the central matrix is not an H-matrix). This shows that for this matrix other usual conditions of P-matrices fail, while we might recognize it as a P-matrix due to it being a B_π^R-matrix. This shows a reason for studying this matrix class.

Now, let us conclude this illustration by showing the characterization through reduction on this example. The three instances from the reduction from Proposition 3.1 are:

$$A_1 = \begin{pmatrix} 7.95 & -7.05 & -0.05 \\ 1.05 & 0.95 & 0.95 \\ 9.05 & 10.95 & 19.95 \end{pmatrix}, \; A_2 = \begin{pmatrix} 7.95 & -6.95 & -0.05 \\ 0.95 & 0.95 & 0.95 \\ 8.95 & 11.05 & 19.95 \end{pmatrix},$$
$$A_3 = \begin{pmatrix} 7.95 & -7.05 & 0.05 \\ 0.95 & 0.95 & 1.05 \\ 8.95 & 10.95 & 19.95 \end{pmatrix}.$$

3.2 *B*-matrices

As written at the beginning of this section, we need to use a characterization of interval B-matrices introduced in [10] plus a definition of real B-matrices and one of their characterizations introduced by Peña in [14], so let us state them here.

Definition 3.1 (B-matrix, [14])**.** *Let $A \in \mathbb{R}^{n\times n}$. We say that A is a* B-matrix *if $\forall i \in [n]$ the following holds:*

$$a) \quad \sum_{j=1}^{n} a_{ij} > 0$$
$$b) \quad \forall k \in [n] \setminus \{i\}: \quad \frac{1}{n}\sum_{j=1}^{n} a_{ij} > a_{ik}$$

Remark 3.3. We can see that from Definition 3.1 we have that B-matrices are B-$\frac{R}{\pi}$matrices for $\pi = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)$. Therefore, the B-$\frac{R}{\pi}$matrices might be seen as a generalization of the B-matrices.

Proposition 3.3. *If $A \in \mathbb{R}^{n\times n}$, then A is a B-matrix if and only if $\forall i \in [n]$ the following holds:*

$$\sum_{j=1}^{n} a_{ij} > n \cdot r_i^+,$$

where $r_i^+ = \max\{0, a_{ij} |\ j \neq i\}$.

Definition 3.2 (Interval B-matrix)**.** *Let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$. We say that $\boldsymbol{A}$ is an* interval B-matrix *if $\forall A \in \boldsymbol{A}$: A is a (real) B-matrix.*

Proposition 3.4. *If $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$, then $\boldsymbol{A}$ is an interval B-matrix if and only if $\forall i \in [n]$ the following two properties hold:*

$$a) \quad \sum_{j=1}^{n} \underline{a}_{ij} > 0$$
$$b) \quad \forall k \in [n] \setminus \{i\}: \quad \sum_{j\neq k} \underline{a}_{ij} > (n-1) \cdot \overline{a}_{ik}$$

Now, let us introduce the reduction.

Proposition 3.5. *Let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$, and let $\forall i \in [n] : A_i$ be matrices defined as follows:*

$$A_i = (a_{m_1m_2}); \quad a_{m_1m_2} = \begin{cases} \overline{a}_{m_1m_2} & \text{if } m_1 \neq i, m_2 = i, \\ \underline{a}_{m_1m_2} & \text{otherwise.} \end{cases}$$

It follows that $\boldsymbol{A}$ is an interval B-matrix if and only if $\forall i \in [n] : A_i$ is a B-matrix.

Proof. "⇒" This holds trivially, because $\forall i \in [n] : A_i \in \boldsymbol{A}$

"⇐"

a)$\forall i \in [n] : \sum_{j=1}^{n} \underline{a}_{ij} > 0$, because A_i is a B-matrix, and $(A_i)_{i,*} = (\underline{A})_{i,*}$, so the row sums of $\underline{A}$ are positive.

b)$\forall i \in [n]\ \forall k \neq i :$ A_k is a B-matrix ⇒ (From Proposition 3.3:)

$$\overline{a}_{ik} + \sum_{j\neq k} \underline{a}_{ij} \quad = \quad \sum_{j=1}^{n} (A_k)_{ij} \quad > \quad n \cdot r_i^+ \quad \geq \quad n \cdot (A_k)_{ik} \quad = \quad n \cdot \overline{a}_{ik}$$

⇒

$$\sum_{j\neq k} \underline{a}_{ij} \quad > \quad (n-1) \cdot \overline{a}_{ik}$$

Whence it follows that $\boldsymbol{A}$ fulfills the conditions of Proposition 3.4, and so is an interval B-matrix. □

Proposition 3.6. *The characterization of interval B-matrices through the reduction given by Proposition 3.5 is minimal with respect to inclusion.*

Proof. If we skip any A_i for arbitrary $i \in [n]$, then we would be able to construct a counterexample, e.g., a unit matrix with an additional interval $[0, 1]$ on position (j, i) for arbitrary $j \neq i$. Then $\forall k \neq i : A_k = I_n$, which surely is a B-matrix, but A_i does not fulfill condition *b*) from Definition 3.1 in the j-th row. (The sum of the j-th row is equal to 2, so we get $2/n > 1 = (A_i)_{ji}$, which does not hold for $n \geq 2$.) □

Remark 3.4. This reduction reduces the problem of verifying whether any given interval matrix is an interval B-matrix, into testing whether n matrices are real B-matrices.

3.3 Doubly *B*-matrices

As written at the beginning of this section, we need to use a characterization of interval doubly B-matrices introduced in [10] and a definition of real doubly B-matrices introduced by Peña in [15], so let us state them here.

Definition 3.3 (Doubly B-matrix, [15])**.** *Let $A \in \mathbb{R}^{n\times n}$. We say that A is a* doubly B-matrix *if $\forall i \in [n]$ the following holds:*

a) $a_{ii} > r_i^+$

b) $\forall j \in [n] \setminus \{i\} : \left(a_{ii} - r_i^+\right)\left(a_{jj} - r_j^+\right) > \left(\sum_{k\neq i} \left(r_i^+ - a_{ik}\right)\right)\left(\sum_{k\neq j} \left(r_j^+ - a_{jk}\right)\right)$

Remark 3.5. We can rearrange the inequality from Proposition 3.3 and hence obtain the following characterization of B-matrices:

$$\forall i \in [n] : \left(a_{ii} - r_i^+\right) > \sum_{k \neq i} \left(r_i^+ - a_{ik}\right)$$

This shows that doubly B-matrices are another generalization of B-matrices. Is there then any difference between the two generalizations, between doubly B-matrices and B^R_π-matrices? Yes, there is. The two matrix classes indeed have a nonempty intersection with B-matrices in it, however, as we will see in the following example, the intersection is just a proper subset of each of those two classes.

Example 3.2. Let us show two examples of matrices that demonstrate the difference between the class of doubly B-matrices and that of B^R_π-matrices.

$$\begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

✓ Doubly B-matrix	× Doubly B-matrix
× B^R_π-matrix (for no π)	✓ B^R_π-matrix (e.g. $\pi = (\frac{1}{3}, \frac{2}{3})$)
It does not have a positive row sum.	The diagonal element is not the largest.

Definition 3.4 (Interval doubly B-matrix)**.** *Let $\boldsymbol{A} \in \mathbb{IR}^{n \times n}$. We say that $\boldsymbol{A}$ is an* interval doubly B-matrix *if $\forall A \in \boldsymbol{A}$: A is a (real) doubly B-matrix.*

Proposition 3.7. *If $\boldsymbol{A} \in \mathbb{IR}^{n \times n}$, then $\boldsymbol{A}$ is an interval doubly B-matrix if and only if the following two properties hold:*

a) $\forall i \in [n] : \quad \underline{a}_{ii} > \max\{0, \overline{a}_{ij} | j \neq i\}$, *and*

b) $\forall (i,j) \in [n]^2, j \neq i, \forall (k,l) \in [n]^2, k \neq i, l \neq j :$

I.
$$\left(\underline{a}_{ii} - \overline{a}_{ik}\right)\left(\underline{a}_{jj} - \overline{a}_{jl}\right) > \left(\max\left\{0, \sum_{\substack{m \neq i \\ m \neq k}} \left(\overline{a}_{ik} - \underline{a}_{im}\right)\right\}\right)\left(\max\left\{0, \sum_{\substack{m \neq j \\ m \neq l}} \left(\overline{a}_{jl} - \underline{a}_{jm}\right)\right\}\right)$$

II.
$$\underline{a}_{ii}\left(\underline{a}_{jj} - \overline{a}_{jl}\right) > \left(\max\left\{0, -\sum_{m \neq i} \underline{a}_{im}\right\}\right)\left(\max\left\{0, \sum_{\substack{m \neq j \\ m \neq l}} \left(\overline{a}_{jl} - \underline{a}_{jm}\right)\right\}\right)$$

III.
$$\underline{a}_{ii} \cdot \underline{a}_{jj} > \left(\max\left\{0, -\sum_{m \neq i} \underline{a}_{im}\right\}\right)\left(\max\left\{0, -\sum_{m \neq j} \underline{a}_{jm}\right\}\right)$$

Now, let us present the reductions.

Proposition 3.8. *Let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ for $n \geq 4$, and let us define $A_{(i,k),(j,l)} \in \mathbb{R}^{n\times n}$ as follows:*

$$A_{(i,k),(j,l)} = (a_{m_1m_2}); \quad a_{m_1m_2} = \begin{cases} \overline{a}_{ik} & \text{if } (m_1,m_2) = (i,k), \\ \overline{a}_{jl} & \text{if } (m_1,m_2) = (j,l), \\ \underline{a}_{m_1m_2} & \text{otherwise.} \end{cases}$$

It holds that $\boldsymbol{A}$ is an interval doubly B-matrix if and only if $\forall (i,j) \in [n]^2, j > i, \forall (k,l) \in [n]^2, k \neq i, l \neq j : A_{(i,k),(j,l)}$ is a doubly B-matrix.

Proof. "$\Rightarrow$" Trivial, for all such matrices: $A_{(i,k),(j,l)} \in \boldsymbol{A}$.

"$\Leftarrow$" We prove that the conditions of Proposition 3.7 hold:

a) $\forall i \in [n], \forall k \neq i : \quad \underline{a}_{ii} > \max\{0, \overline{a}_{ik}\}$, because for any arbitrary j, l the matrix $A_{(i,k),(j,l)}$ is a doubly B-matrix. Hence, $\forall i \in [n] : \quad \underline{a}_{ii} > \max\{0, \overline{a}_{ik} | k \neq i\}$.

b) Let us fix arbitrary $(i,j) \in [n]^2, j \neq i$ and arbitrary $(k,l) \in [n]^2, k \neq i, l \neq j$. Without loss of generality suppose $j > i$. (If $j < i$, we swap their values and we also swap the values of k and l, too.) Let us define $A = A_{(i,k),(j,l)}$ to simplify notation. Then:

I.

$$\begin{aligned} & \left(\underline{a}_{ii} - \overline{a}_{ik}\right)\left(\underline{a}_{jj} - \overline{a}_{jl}\right) \quad \geq \quad \left(a_{ii} - r_i^+\right)\left(a_{jj} - r_j^+\right) \\ > \quad & \left(\sum_{m\neq i} \left(r_i^+ - a_{im}\right)\right)\left(\sum_{m\neq j} \left(r_j^+ - a_{jm}\right)\right) \\ \geq \quad & \left(\max\left\{0, \sum_{\substack{m\neq i \\ m\neq k}} \left(\overline{a}_{ik} - \underline{a}_{im}\right)\right\}\right)\left(\max\left\{0, \sum_{\substack{m\neq j \\ m\neq l}} \left(\overline{a}_{jl} - \underline{a}_{jm}\right)\right\}\right) \end{aligned}$$

The second inequality holds, because A is a doubly B-matrix.

II.

$$\begin{aligned} & \underline{a}_{ii}\left(\underline{a}_{jj} - \overline{a}_{jl}\right) \quad \geq \quad \left(a_{ii} - r_i^+\right)\left(a_{jj} - r_j^+\right) \\ > \quad & \left(\sum_{m\neq i} \left(r_i^+ - a_{im}\right)\right)\left(\sum_{m\neq j} \left(r_j^+ - a_{jm}\right)\right) \\ \geq \quad & \left(\max\left\{0, -\sum_{m\neq i} \underline{a}_{im}\right\}\right)\left(\max\left\{0, \sum_{\substack{m\neq j \\ m\neq l}} \left(\overline{a}_{jl} - \underline{a}_{jm}\right)\right\}\right) \end{aligned}$$

The second inequality holds because of the fact that $A_{(x,y),(j,l)}$ for any $x \neq i$ and $y \neq x$ is a doubly B-matrix.

III.

$$\begin{aligned} \underline{a}_{ii} \cdot \underline{a}_{jj} &\geq \left(a_{ii} - r_i^+\right)\left(a_{jj} - r_j^+\right) \\ &> \left(\sum_{m\neq i}\left(r_i^+ - a_{im}\right)\right)\left(\sum_{m\neq j}\left(r_j^+ - a_{jm}\right)\right) \\ &\geq \left(\max\left\{0, -\sum_{m\neq i}\underline{a}_{im}\right\}\right)\left(\max\left\{0, -\sum_{m\neq j}\underline{a}_{jm}\right\}\right) \end{aligned}$$

The second inequality holds because of the fact that $A_{(x,y),(u,v)}$ for any x, y, u, v, such that $x \neq i, x \neq j, y \neq x, u \neq i, u \neq j, u \neq x$, and $v \neq u$ is a doubly B-matrix and $n \geq 4$.

Thus, as we have shown, the $\boldsymbol{A}$ fulfills both the conditions of Proposition 3.7, therefore it is an interval doubly B-matrix. □

Remark 3.6. Proposition 3.8 could also work for $n \geq 3$, but we would have to add a requirement that $\underline{A}$ is a doubly B-matrix, too. Or it could work even for $n \geq 2$, but again we would have to add requirements that $\underline{A}$ is a doubly B-matrix and $\forall j \in [n], l \neq j : A_{(j,l)}$ is a doubly B-matrix, where

$$A_{(j,l)} = (a_{m_1 m_2}); \quad \begin{cases} \overline{a}_{jl} & \text{if } (m_1, m_2) = (j, l), \\ \underline{a}_{m_1 m_2} & \text{otherwise.} \end{cases}$$

These requirements are needed for proof of parts "*II.*" and "*III.*" of condition *b)* of the second (right-to-left) implication. However, we can show an example that they are not just formal requirements:

Example 3.3. Let $\boldsymbol{A} \in \mathbb{IR}^{3\times 3}$, such that $\boldsymbol{A}_{ij} = \begin{cases} [1,1] = 1 & \text{if } i = j, \\ [-\frac{1}{2}, 0] & \text{otherwise.} \end{cases}$

Then $\forall A_{(i,k),(j,l)}$: $\forall z, z' \in [3], z' \neq z : r_z^+ = r_{z'}^+ = 0$, so:

$$\left(a_{zz} - r_z^+\right)\left(a_{z'z'} - r_{z'}^+\right) = 1 \cdot 1 = 1,$$

and

$$\left(\sum_{m\neq z}\left(r_z^+ - a_{zm}\right)\right)\left(\sum_{m\neq z'}\left(r_{z'}^+ - a_{z'm}\right)\right) \leq \frac{1}{2} \cdot 1 = \frac{1}{2}.$$

Thus, every $A_{(i,k),(j,l)}$ is a doubly B-matrix.

However, for $\underline{A}$: $\forall z, z' \in [3], z' \neq z$:

$$\left(a_{zz} - r_z^+\right)\left(a_{z'z'} - r_{z'}^+\right) = 1 \cdot 1 = 1,$$

and

$$\left(\sum_{m\neq z}\left(r_z^+ - a_{zm}\right)\right)\left(\sum_{m\neq z'}\left(r_{z'}^+ - a_{z'm}\right)\right) = \left(\frac{1}{2} + \frac{1}{2}\right)^2 = 1^2 = 1.$$

Therefore, $\underline{A}$ is not a doubly B-matrix, and hence $\boldsymbol{A}$ cannot be an interval doubly B-matrix.

Proposition 3.9. *The characterization of interval doubly B-matrices through the reduction given by Proposition 3.8 is minimal with respect to inclusion.*

Proof. If we skip $A_{(i,k),(j,l)}$ for any arbitrary $(i,j,k,l) \in [n]^4, j \neq i, k \neq i, l \neq j$, then we would be able to construct a counterexample, e.g., a unit matrix with an additional interval $[0, \frac{1}{2}]$ at positions (i,k) and (j,l). Then $\forall (x,y,u,v) \in [n]^4, u \neq x, y \neq x, v \neq u$, such that $(x,y,u,v) \neq (i,k,j,l) : A_{(x,y),(u,v)}$ is a doubly B-matrix. That holds because $\forall (z,z') \in [n]^2, z' \neq z$:

$$\left(a_{zz} - r_z^+\right)\left(a_{z'z'} - r_{z'}^+\right) \geq \frac{1}{2},$$

and

$$\left(\sum_{m \neq z} \left(r_z^+ - a_{zm}\right)\right)\left(\sum_{m \neq z'} \left(r_{z'}^+ - a_{z'm}\right)\right) = 0.$$

However, $A_{(i,k),(j,l)}$ is not a doubly B-matrix, because

$$\left(a_{ii} - r_i^+\right)\left(a_{jj} - r_j^+\right) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4},$$

and

$$\left(\sum_{m \neq i} \left(r_i^+ - a_{im}\right)\right)\left(\sum_{m \neq j} \left(r_j^+ - a_{jm}\right)\right) = \left(\left(\frac{1}{2} - \frac{1}{2}\right) + (n-2) \cdot \left(\frac{1}{2} - 0\right)\right)^2 = \left(\frac{n-2}{2}\right)^2,$$

and for $n \geq 3$ it does not hold that $\frac{1}{4} > \left(\frac{n-2}{2}\right)^2$. (Plus in Proposition 3.8 we assume $n \geq 4$.)

Hence, the whole interval matrix cannot be an interval doubly B-matrix. □

Whereas the previous reduction stated in Proposition 3.8 reduces the problem of verifying an interval matrix on being an interval doubly B-matrix to $O(n^4)$ matrices (more precisely, for its basic version for $n \geq 4$ it reduces the problem to $\binom{n}{2} \cdot (n-1)^2$ real instances), the following uses a bit different approach and achieves to reduce the definition to $O(n^3)$ (more precisely to $n^2 \cdot (n-1) + n^2 = n^3$) matrices.

Proposition 3.10. *Let $\boldsymbol{A} \in \mathbb{IR}^{n \times n}$, and let us define $A_{(i,k),(*,l)}$ and ${}_iA_{(*,l)} \in \mathbb{R}^{n \times n}$ as follows:*

$$A_{(i,k),(*,l)} = (a_{m_1 m_2}); \quad a_{m_1 m_2} = \begin{cases} \overline{a}_{ik} & \text{if } (m_1, m_2) = (i,k), \\ \overline{a}_{m_1 l} & \text{if } m_2 = l \wedge m_1 \neq i \wedge m_1 \neq l, \\ \underline{a}_{m_1 m_2} & \text{otherwise.} \end{cases}$$

and

$${}_iA_{(*,l)} = \left(a'_{m_1 m_2}\right); \quad a'_{m_1 m_2} = \begin{cases} \overline{a}_{m_1 l} & \text{if } m_2 = l \wedge m_1 \neq i \wedge m_1 \neq l, \\ \underline{a}_{m_1 m_2} & \text{otherwise.} \end{cases}$$

The matrix $\boldsymbol{A}$ *is an interval doubly B-matrix if and only if* $\forall (i,l) \in [n]^2 : ({}_iA_{(*,l)}$ *is a doubly B-matrix* $\wedge \quad \forall k \in [n] \setminus \{i\} : A_{(i,k),(*,l)}$ *is a doubly B-matrix).*

Proof. "$\Rightarrow$" Trivial, for all such matrices are in $\boldsymbol{A}$.

"$\Leftarrow$" We prove that the conditions of Proposition 3.7 hold:

a) $\forall i \in [n], \forall k \neq i : \quad \underline{a}_{ii} > \max\{0, \overline{a}_{ik}\}$, because for any arbitrary l the matrix $A_{(i,k),(*,l)}$ is a doubly B-matrix. Therefore, $\forall i \in [n] : \quad \underline{a}_{ii} > \max\{0, \overline{a}_{ik} | k \neq i\}$.

b) Let us fix arbitrary $(i,j) \in [n]^2, j \neq i$ and arbitrary $(k,l) \in [n]^2, k \neq i, l \neq j$.

I. Let us take $A = A_{(i,k),(*,l)}$. Then:

$$\begin{aligned}
&\left(\underline{a}_{ii} - \overline{a}_{ik}\right)\left(\underline{a}_{jj} - \overline{a}_{jl}\right) \quad \geq \quad \left(a_{ii} - r_i^+\right)\left(a_{jj} - r_j^+\right) \\
> \quad &\left(\sum_{m \neq i} \left(r_i^+ - a_{im}\right)\right)\left(\sum_{m \neq j} \left(r_j^+ - a_{jm}\right)\right) \\
\geq \quad &\left(\max\left\{0, \sum_{\substack{m \neq i \\ m \neq k}} \left(\overline{a}_{ik} - \underline{a}_{im}\right)\right\}\right)\left(\max\left\{0, \sum_{\substack{m \neq j \\ m \neq l}} \left(\overline{a}_{jl} - \underline{a}_{jm}\right)\right\}\right)
\end{aligned}$$

II. Let us take $A = {}_iA_{(*,l)}$. Then:

$$\begin{aligned}
&\underline{a}_{ii}\left(\underline{a}_{jj} - \overline{a}_{jl}\right) \quad \geq \quad \left(a_{ii} - r_i^+\right)\left(a_{jj} - r_j^+\right) \\
> \quad &\left(\sum_{m \neq i} \left(r_i^+ - a_{im}\right)\right)\left(\sum_{m \neq j} \left(r_j^+ - a_{jm}\right)\right) \\
\geq \quad &\left(\max\left\{0, -\sum_{m \neq i} \underline{a}_{im}\right\}\right)\left(\max\left\{0, \sum_{\substack{m \neq j \\ m \neq l}} \left(\overline{a}_{jl} - \underline{a}_{jm}\right)\right\}\right)
\end{aligned}$$

III. Let us take $A = {}_iA_{(*,j)}$. Then:

$$\begin{aligned}
&\underline{a}_{ii} \cdot \underline{a}_{jj} \quad \geq \quad \left(a_{ii} - r_i^+\right)\left(a_{jj} - r_j^+\right) \\
> \quad &\left(\sum_{m \neq i} \left(r_i^+ - a_{im}\right)\right)\left(\sum_{m \neq j} \left(r_j^+ - a_{jm}\right)\right) \\
\geq \quad &\left(\max\left\{0, -\sum_{m \neq i} \underline{a}_{im}\right\}\right)\left(\max\left\{0, -\sum_{m \neq j} \underline{a}_{jm}\right\}\right)
\end{aligned}$$

Therefore, as we have proved, the $\boldsymbol{A}$ fulfills both the conditions of characterization stated in Proposition 3.7, thus it is an interval doubly B-matrix. □

4 Conclusion and future work

There are several ways in which the current results might be extended. One possibility is to generalize our three classes even further, into parametric matrices, otherwise known as linearly dependent, addressed, for example, in [17]. Another direction is to generalize another subclass of P-matrices. Those might be, for example, so-called mimes, which stands for "M-matrix and Inverse M-matrix Extension", as they were introduced in [18]. Or it still remains unresolved whether the reductions presented in this paper are optimal with respect to the number of real instances used, or whether there exists some other reduction achieving to characterize one of the interval matrix classes using fewer instances. For the reduction from Proposition 3.10, the minimality with respect to inclusion is still undecided.

References

[1] Cottle, Richard W., Pang, Jong-Shi, and Stone, Richard E. *The Linear Complementarity Problem*. SIAM, Philadelphia, PA, revised ed. of the 1992 original edition, 2009. DOI: 10.1137/1.9780898719000.

[2] Coxson, Gregory E. The P-matrix problem is co-NP-complete. *Math. Program.*, 64(1):173–178, 1994. DOI: 10.1007/BF01582570.

[3] Garloff, Jürgen, Adm, Mohammad, and Titi, Jihad. A survey of classes of matrices possessing the interval property and related properties. *Reliab. Comput.*, 22:1–10, 2016. URL: https://www.reliable-computing.org/reliable-computing-22-pp-001-014.pdf.

[4] Garloff, Jürgen, Al-Saafin, Doaa, and Adm, Mohammad. Further matrix classes possessing the interval property. *Reliab. Comput.*, 28:56–70, 2021. URL: https://www.reliable-computing.org/reliable-computing-28-pp-056-070.pdf.

[5] Hladík, Milan. On relation between P-matrices and regularity of interval matrices. In Bebiano, Natália, editor, *Applied and Computational Matrix Analysis*, volume 192 of *Springer Proceedings in Mathematics & Statistics*, pages 27–35. Springer, 2017. DOI: 10.1007/978-3-319-49984-0_2.

[6] Hladík, Milan. An overview of polynomially computable characteristics of special interval matrices. In Kosheleva O. et al, editor, *Beyond Traditional Probabilistic Data Processing Techniques: Interval, Fuzzy etc. Methods and Their Applications*, volume 835 of *Studies in Computational Intelligence*, pages 295–310. Springer, Cham, 2020. DOI: 10.1007/978-3-030-31041-7_16.

[7] Hogben, Leslie, editor. *Handbook of Linear Algebra*. Chapman & Hall/CRC, 2007.

[8] Horáček, Jaroslav, Hladík, Milan, and Černý, Michal. Interval linear algebra and computational complexity. In Bebiano, Natália, editor, *Applied and Computational Matrix Analysis*, volume 192 of *Springer Proceedings in Mathematics & Statistics*, pages 37–66. Springer, 2017. DOI: 10.1007/978-3-319-49984-0_3.

[9] Kreinovich, Vladik, Lakeyev, Anatoly, Rohn, Jiří, and Kahl, Patrick. *Computational Complexity and Feasibility of Data Processing and Interval Computations*. Kluwer, Dordrecht, 1998. DOI: 10.1007/978-1-4757-2793-7.

[10] Lorenc, Matyáš. Special classes of P-matrices in the interval setting. Bachelor's thesis, Department of Applied Mathematics, Charles University, 2021.

[11] Mendes Araújo, C. and Mendes-Gonçalves, S. On a class of nonsingular matrices containing B-matrices. *Linear Algebra Appl.*, 578:356–369, 2019. DOI: 10.1016/j.laa.2019.05.015.

[12] Neumann, Michael, Peña, J.M., and Pryporova, Olga. Some classes of nonsingular matrices and applications. *Linear Algebra Appl.*, 438(4):1936–1945, 2013. DOI: 10.1016/j.laa.2011.10.041.

[13] Orera, Héctor and Peña, Juan Manuel. Error bounds for linear complementarity problems of B_π^R-matrices. *Comput. Appl. Math.*, 40(3):94:1–94:13, 2021. DOI: 10.1007/s40314-021-01491-w.

[14] Peña, J. M. A class of P-matrices with applications to the localization of the eigenvalues of a real matrix. *SIAM J. Matrix Anal. Appl.*, 22(4):1027–1037, 2001. DOI: 10.1137/S0895479800370342.

[15] Peña, J. M. On an alternative to Gerschgorin circles and ovals of Cassini. *Numer. Math.*, 95(2):337–345, 2003. DOI: 10.1007/s00211-002-0427-8.

[16] Rohn, Jiří. On Rump's characterization of P-matrices. *Optim. Lett.*, 6(5):1017–1020, 2012. DOI: 10.1007/s11590-011-0318-y.

[17] Skalna, Iwona. *Parametric Interval Algebraic Systems*, volume 766 of *Studies in Computational Intelligence*. Springer, Cham, 2018. DOI: 10.1007/978-3-319-75187-0.

[18] Tsatsomeros, Michael J. Generating and detecting matrices with positive principal minors. In Li, Lei, editor, *Focus on Computational Neurobiology*, pages 115–132. Nova Science Publishers, Commack, NY, USA, 2004.

Acta Cybernetica **26** (2023) 105–128.

Quantification of Time-Domain Truncation Errors for the Reinitialization of Fractional Integrators

Andreas Rauh[a] and Rachid Malti[b]

Abstract

In recent years, fractional differential equations have received a significant increase in their use for modeling a wide range of engineering applications. In such cases, they are mostly employed to represent non-standard dynamics that involve long-term memory effects or to represent the dynamics of system models that are identified from measured frequency response data in which magnitude and phase variations are observed that could be captured either by low-order fractional models or high-order rational ones. Fractional models arise also when synthesizing CRONE (Commande Robuste d'Ordre Non Entier) and/or fractional PID controllers for rational or fractional systems. In all these applications, it is frequently required to transform the frequency domain representation into time domain. When doing so, it is necessary to carefully address the issue of the initialization of the pseudo state variables of the time domain system model. This issue is discussed in this article for the reinitialization of fractional integrators which arises among others when solving state estimation tasks for continuous-time systems with discrete-time measurements. To quantify the arising time-domain truncation errors due to integrator resets, a novel interval observer-based approach is presented and, finally, visualized for a simplified battery model.

Keywords: fractional differential equations (FDEs), observer design, uncertain cooperative dynamics, temporal truncation errors, state estimation

1 Introduction

Fractional differential equations (FDEs) are powerful modeling tools in many engineering applications in which non-standard dynamics, characterized by infinite horizon states, can be observed [21, 23, 37, 40]. Examples for such applications are modeling the charging and discharging dynamics of batteries [11], the identification of dynamic system models by means of impedance spectroscopy [2] if amplitude

[a]Carl von Ossietzky Universität Oldenburg, Department of Computing Science, Group: Distributed Control in Interconnected Systems, D-26111 Oldenburg, Germany, E-mail: Andreas.Rauh@uni-oldenburg.de, ORCID: 0000-0002-1548-6547

[b]IMS Laboratory, University of Bordeaux, 33405 Talence, France, E-mail: Rachid.Malti@u-bordeaux.fr, ORCID: 0000-0002-5026-9919

DOI: 10.14232/actacyb.296010

and phase variations do not correspond to integer multiples of $\pm 20\,\mathrm{dB}$ and $\pm\frac{\pi}{2}$ per frequency decade, respectively, modeling of multi-robot systems [9], control design for flexible manipulators [4], and generally for the representation of dynamic systems with long-term memory effects. Moreover, advanced models of visco-elastic damping [13] can be described with the help of FDEs. In this domain, FDEs can be used to describe phenomena with non-integer time derivatives, which represent phenomena lying "between" Hooke's law (with a proportionality of forces to the displacement) and Newton's law (being proportional to the first time derivative of the displacement) [16]. Similar effects also exist in the domain of heat and mass transport, where non-standard dynamics may be related to phenomena "between" diffusion and wave propagation [36].

Previous work for an interval-based state estimation of such systems has accounted for a cooperativity preserving or cooperativity enforcing design of observers [3, 11]. These interval observers exploit specific monotonicity properties of positive dynamic systems and provide lower and upper bounding trajectories for all pseudo state variables[1] as soon as suitable initialization functions for the fractional dynamic system model are specified.

Moreover, FDEs arise naturally if the CRONE design methodology [15,22] for efficient shaping of frequency response characteristics of linear control systems and/or if fractional PID controllers [20, 24] are employed. FDEs then arise independently of whether the plant to be controlled is represented by a rational or fractional system model, and if the resulting closed-loop system dynamics are subsequently represented in the time domain.

In contrast to the case of integer-order models, the time responses of fractional systems significantly depend on the initialization of the pseudo state. This is shown exemplarily in this paper with the help of the Grünwald-Letnikov definition of fractional derivatives [23, 32, 34, 39] to illustrate further that the Caputo initialization corresponds to the special case that an FDE model is initialized with an initial condition that also represents a perfectly constant, infinitely long history of the pseudo states in the past. Although this may hold (at least in good approximation) for the initialization of a dynamic system which is fully in rest, this is obviously not true when resetting integrators after a finitely long time interval.

Apart from the discussions above, interval-valued iteration procedures have been developed in [26–29] for a verified simulation of FDE models. These iteration procedures, based on Mittag-Leffer function parameterizations of the pseudo-state enclosures, are not *a priori* restricted to cooperative models but are applicable also to nonlinear systems with interval parameters. So far, this procedure assumes that — for the initialization — a fractional derivative definition according to Caputo is used. This verified simulation, however, allows for resetting the integration after a finite time span by applying (to our knowledge, for the first time in a verified simulation of FDEs) an error quantification originally published in the book [23]

[1] The notion *pseudo state* is used throughout this manuscript to indicate the existence of the infinite memory problem of FDEs in contrast to the classical notion of *state variables* that only need to be specified at distinct points in time to unambiguously solve initial or boundary value problems for classical integer-order dynamic system models.

by Podlubny.

In this paper, we aim at improving this error quantification scheme by a novel interval observer-based approach that allows for estimating guaranteed interval bounds for time-domain truncation errors in scenarios in which fractional integrators need to be reset. Such cases occur when state estimation for continuous-time FDE models with discrete-time measurements is considered. So far, the state-of-the-art in the evaluation of observer-based pseudo-state estimation procedures for continuous-time FDE models supposes that measurements are also available in a continuous-time form or at least at each sampling period [3]. If measurements are available only at discrete time instants, continuous-time pseudo-state predictions need to be performed between the sampling instants of the measurements.

Then, the measured pseudo-state information (described by intervals to represent bounded measurement errors) can be intersected with the predicted state information to enhance the knowledge of the actual system dynamics. However, this intersection demands reinitializing the integration of the fractional model. As already mentioned above, a similar requirement is discussed in [26], where temporal sub-slices were considered to reduce the overestimation of interval-based simulation approaches. Moreover, such integrator resets also help to limit memory demands that would grow continuously with increasing integration times if solutions to FDEs were approximated with the help of series expansion techniques based on the Grünwald-Letnikow definition of fractional derivatives.

Due to the infinite horizon memory property of fractional systems, the reinitialization of time-domain simulations requires a rigorous consideration of the arising truncation errors. Although guaranteed outer bounds for these errors were derived by Podlubny in [23], they may be unnecessarily conservative due to an assumption of the time invariance of these bounds for all future points after the integrator reset. We aim at using Podlubny's initial bounds as a basis for a novel error refinement strategy between discrete reinitialization points in an observer-based setting.

After an introduction into the infinite memory problem of FDE models in Sec. 2, an approach that accounts for handling non-constant pseudo-state initializations from a bounded past time window in terms of uncertain initial conditions at a single point is derived. This approach is based on a conservative interval-valued correction of the FDE model. It forms the basis for implementing an observer-based quantification of truncation errors for simulations of FDEs in which a periodic reinitialization is employed in Sec. 3. This approach is then applied in Sec. 4 to an academic benchmark example as well as to the interval contractor-based state estimation of a continuous-time battery model [11] with discrete-time measurements, before the paper is concluded with an outlook on future work in Sec. 5.

2 Influence of the Initialization of FDE Models

To visualize the influence of the initialization of the pseudo state of FDE models on their future behavior, consider the representation of the solution of a commensurate

autonomous FDE

$$\mathbf{x}^{(\nu)}(t) = \mathbf{f}\left(\mathbf{x}(t)\right) \quad , \quad \mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^n, \tag{1}$$

in terms of the infinite series

$$\mathbf{x}(t_{k+1}) = \nu\mathbf{I} \cdot \mathbf{x}(t_k) + \Delta T_k^{\nu} \cdot \mathbf{f}\left(\mathbf{x}(t_k)\right) - \sum_{i=2}^{\infty} c_i \cdot \mathbf{x}(t_{k+1-i}) \tag{2}$$

with the sufficiently short step size $\Delta T_k = t_{k+1} - t_k$. This series expansion results from the Grünwald-Letnikov definition[2] of a non-integer derivative of order $0 < \nu \leq 1$ with the coefficients

$$c_i = (-1)^i \cdot \binom{\nu}{i} = (-1)^i \cdot \frac{\Gamma(\nu+1)}{\Gamma(i+1) \cdot \Gamma(\nu-i+1)}, \tag{3}$$

in which the term $\binom{\nu}{i}$ is the Newton binomial coefficient with the gamma function

$$\Gamma(\nu) = \int_0^{\infty} \xi^{\nu-1} e^{-\xi} \mathrm{d}\xi \tag{4}$$

as a generalization of the factorial to the case of non-integer arguments. To avoid excessive numerical errors when evaluating the coefficients c_i, they are typically computed in a recursive manner according to [39]

$$c_i = c_{i-1} \cdot \left(1 - \frac{1+\nu}{i}\right) \quad \text{with} \quad c_0 = 1 \quad \text{and} \quad i \in \mathbb{N}. \tag{5}$$

As it can be seen already in Eq. (2), future pseudo states $\mathbf{x}(t_{k+1})$ do not only depend on the current state $\mathbf{x}(t_k)$ as a kind of initialization (as it would be the case for integer-order system models), but they also depend on an infinite horizon of pseudo states from previous points of time $t < t_k$. Note, stability properties of this series expansion and properties of its convergence toward the true solution of a fractional system model have been analyzed in detail in [32]. In fact, the Grünwald-Letnikov discretization can be interpreted as a generalization of the well-known Euler discretization scheme for integer-order models so that the true state evolution can be approximated accurately for sufficiently small values of ΔT_k. For methods that allow a rigorous quantification of time discretization errors, the reader is referred to [27–29], where an exponential state enclosure technique is generalized to fractional models by using an iteration scheme exploiting an interval extension of Mittag-Leffler functions [10,19], or to [1,18,26] where series expansion approaches and Picard iteration schemes were generalized to the fractional case.

For linear FDEs of Caputo type, typically only initial conditions $\mathbf{x}(t_k)$ are specified explicitly at a point of time t_0 that is set to $t_0 = 0$ without loss of generality in the remainder of this paper. As shown in the following example, this specification

[2] This representation corresponds to the one discussed in [26], except for the correction of a small typo in the quoted previous work of the first author.

implicitly imposes that the pseudo state of the system showed an exactly constant behavior for an infinitely long time window in the past.

To perform this investigation, consider the FDE

$$x^{(0.5)}(t) = -x(t) + u(t) \tag{6}$$

with the pseudo state initialization

$$x(t) = x_0 \quad \text{for} \quad t \leq 0 \tag{7}$$

and the constant external control input

$$u(t) = \begin{cases} 0 & \text{for} \quad t < 0 \\ u_0 & \text{for} \quad t \geq 0. \end{cases} \tag{8}$$

Due to the fact that fractional derivatives of constant values in the Caputo sense are zero, the linear change of variables

$$x(t) = y(t) + u_0 \quad \text{with} \quad x^{(0.5)}(t) = y^{(0.5)}(t) \tag{9}$$

yields the equivalent FDE

$$y^{(0.5)}(t) = -y(t) \tag{10}$$

with

$$y(t) = x_0 - u_0 \quad \text{for} \quad t \leq 0 \tag{11}$$

for which the exact solution is given by

$$y(t) = (x_0 - u_0) \cdot E_{0.5,1}\left(-t^{0.5}\right) \quad \text{for} \quad t \geq 0 \tag{12}$$

being equivalent to

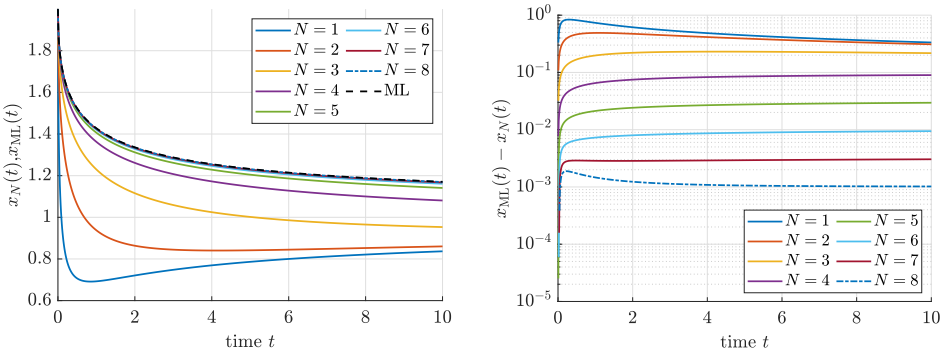
$$x(t) = x_{\mathrm{ML}}(t) = (x_0 - u_0) \cdot E_{0.5,1}\left(-t^{0.5}\right) + u_0 \quad \text{for} \quad t \geq 0. \tag{13}$$

In (12) and (13), $E_{\nu,1}(\cdot)$ is the Mittag-Leffler function with the fractional derivative order $\nu = 0.5$ as parameter.

In Fig. 1, different approximations of the solution to the FDE model (6)–(8) are computed by using the Grünwald-Letnikov approximation with the constant discretization step size $\Delta T_k = 0.01$ and approximations of the infinitely long constant initialization of the pseudo state in (7). These latter approximations are defined by

$$x(t) = x_0 \quad \text{for} \quad t \in \left[-\Delta T_k \cdot 10^N \; ; \; 0\right] \tag{14}$$

with $N \in \{1, 2, \ldots, 8\}$, where the corresponding approximations to the true solution $x(t)$ are denoted by $x_N(t)$. As described above, the exact solution $x_{\mathrm{ML}}(t)$ corresponds to a solution representation in terms of the Mittag-Leffler function according to (13), which has been evaluated in Fig. 1 by the MATLAB implementation by R. Garrappa in [8].



(a) Pseudo state $x(t)$. (b) Approximation error $x_{\mathrm{ML}}(t) - x_N(t)$.

Figure 1: Influence of the memory length of the pseudo state initialization.

It can be seen that insufficiently long memory lengths in the state initialization lead to large deviations between the Grünwald-Letnikov approximation and the true solution. These deviations due to an insufficiently long initialization window are much larger than the influence of the finitely long discretization step size ΔT_k.

In the following section, time domain truncation errors resulting from resetting a numerical integration of an FDE model after a finitely long time span are quantified in a rigorous manner under the assumption that the system behavior for $t \leq 0$ is known in advance. For the sake of simplicity, we rely on the temporally constant initialization according to (7) in the following.

Remark 1. Due to the fact that the numerical solution of an FDE with non-constant initialization functions is influenced by the change of coordinates in (9), this linear shift of the coordinate system will form a potential basis for identifying the history of the pseudo states in future work.

3 Interval Observer Technique for the Identification of Improved Bounds of Time-Domain Truncation Errors

3.1 Constant Bounds for Time-Domain Truncation Errors

So far, we have assumed that initial conditions for the pseudo state of the FDE system model (1) are specified at the instant $t = 0$ with a temporally constant past for all times $t < 0$ in accordance to the Caputo definition of fractional derivatives. To allow for a notation denoting the influence of the point of time at which the derivative operator ${}_0\mathscr{D}_t^{(\nu)}$ is initialized, the notation of Eq. (1) is changed to

$$ {}_0\mathscr{D}_t^{(\nu)}\mathbf{x}(t) = \mathbf{f}\left(\mathbf{x}(t)\right) \tag{15} $$

in the following, where the left subscript of ${}_0\mathscr{D}_t^{(\nu)}$ specifies the initialization point of time (in the case above, $t = 0$).

According to the work of Podlubny [23, Eq. (7.5)], guaranteed bounds for the influence of shifting this initialization point from the time instant t_k to another point $t_k + T$, $T > 0$, can be computed component-wise according to

$$\left| {}_{t_k}\mathscr{D}_t^{(\nu)}\mathbf{x}(t) - {}_{t_k+T}\mathscr{D}_t^{(\nu)}\mathbf{x}(t) \right| \leq \frac{\boldsymbol{\mathscr{X}} T^{-\nu}}{\left|\Gamma(1-\nu)\right|} =: \boldsymbol{\mu} \tag{16}$$

with

$$\boldsymbol{\mathscr{X}} = \begin{bmatrix} \mathscr{X}_1 & \ldots & \mathscr{X}_n \end{bmatrix}^T \tag{17}$$

comprising the suprema

$$\mathscr{X}_i = \sup_{t \in [t_k \,;\, t_k+T]} \left| x_i(t) \right| \;, \quad i \in \{1, \ldots, n\} \tag{18}$$

of the reachable pseudo states over the time interval $t \in [t_k \,;\, t_k + T]$ for each element of the vector $\mathbf{x}(t)$.

As shown in [28], these error bounds can be employed to reset interval-based verified solution procedures for FDE models after a certain time span and to use the solution enclosures determined after the reset to reduce overestimation arising due to pessimism that is introduced by long integration time intervals. For that purpose, the right-hand side of the system model (15) is inflated by the error bound interval $[-\boldsymbol{\mu} \,;\, \boldsymbol{\mu}]$ to obtain the uncertain system model

$$\tilde{\mathbf{f}}\left(\mathbf{x}(t)\right) \in \mathbf{f}\left(\mathbf{x}(t)\right) + [-\boldsymbol{\mu} \,;\, \boldsymbol{\mu}] \,. \tag{19}$$

Using this modification, the simulation is continued after the point $t = t_k + T$ for the differential inclusion model defined by the expression $\tilde{\mathbf{f}}\left(\mathbf{x}(t)\right)$ and the pseudo state values $\mathbf{x}(t_k)$ as initial condition, while the entire past for $t < t_k$ is no longer required for a further system simulation.

Under the assumption of cooperativity of the state equations, see [3,5,25,33] for further details, independent lower and upper bounding trajectories can be extracted from the modified system model (19) so that set-based integration routines such as the one based on interval extensions of the Mittag-Leffler function from [27–29] can be avoided when solving the corresponding initial value problem for the differential inclusion problem (19) after the inflation of the right-hand side $\mathbf{f}\left(\mathbf{x}(t)\right)$ of the original system.

Remark 2. To limit the pessimism introduced by the additive error bounds in (19), the following two aspects should be accounted for:

- Define the pseudo state $\mathbf{x}(t)$ in such a way that $\mathbf{x} = \mathbf{0}$ corresponds to the equilibrium of an asymptotically stable FDE. If $\mathbf{x} = \mathbf{0}$ is not the corresponding steady state after a first-principle modeling, perform a shift of coordinates as inspired by Eq. (11) so that the absolute values for the bounds $\boldsymbol{\mathscr{X}}$ do not increase for sufficiently large values of $t_k + T$.

- Set the initial point t_k in (16) to $t_k = 0$. Together with the first aspect in this remark, this allows for a computation of values for the error bounds $\boldsymbol{\mu}$ that decrease after sufficiently long integration times and thus lead to less conservative system models than always recomputing the bounds $\boldsymbol{\mu}$ with respect to a previous reset point $t_k > 0$.

Even though these two aspects can be accounted for in many practical situations, the bounds $\boldsymbol{\mu}$ given in (16) remain conservative due to the fact that they are temporally constant. This property does not explicitly account for the observation that shifting the initialization point of the fractional derivative operator becomes less important for increasing integration times[3]. Therefore, an observer-based refinement of the bounds $\boldsymbol{\mu}$ — to our knowledge not yet considered in any other publication — is presented in the following subsection. A similar approach, however, can be found in [35], where the authors propose an observer to initialize fractional system models consistently.

3.2 Observer-Based Enhancement of the Bounds for Time-Domain Truncation Errors

For the observer-based enhancement of the time-domain truncation error bounds when resetting fractional integrators, we restrict ourselves to the case of cooperative system models in this paper.

As a generator for *virtual measurements* of a cooperative dynamic system model, we compute pseudo state enclosures

$$\mathbf{x}(t) \in \left[\mathbf{v}(t)\,;\,\mathbf{w}(t)\right] \tag{20}$$

for the FDE model (15) with the temporally constant initialization

$$\mathbf{x}(t) \in [\mathbf{x}_0] \;\;,\;\; \dot{\mathbf{x}}(t) = \mathbf{0} \quad \text{for} \quad t < 0. \tag{21}$$

This setting corresponds to uncertain initial conditions in the sense of Caputo, while the influence of temporally varying initializations is taken into account as soon as the first integrator reset has been performed.

Cooperativity of the system model (15) is guaranteed as a sufficient condition if all off-diagonal elements of the Jacobian of the right-hand side of the system model with respect to the pseudo state vector $\mathbf{x}(t)$ $(i, j \in \{1, \ldots, n\},\ i \neq j)$ satisfy the inequalities

$$\frac{\partial f_i(\mathbf{x})}{\partial x_j} \geq 0. \tag{22}$$

Then, all reachable pseudo states can be enclosed by the lower and upper bounding systems

$$\begin{aligned} {}_0\mathscr{D}_t^{(\nu)}\mathbf{v}(t) &= \mathbf{f}_v\left(\mathbf{v}(t)\right) \;\;,\;\; \mathbf{v}(t \leq 0) = \underline{\mathbf{x}}_0 \quad \text{and} \\ {}_0\mathscr{D}_t^{(\nu)}\mathbf{w}(t) &= \mathbf{f}_w\left(\mathbf{w}(t)\right) \;\;,\;\; \mathbf{w}(t \leq 0) = \overline{\mathbf{x}}_0, \end{aligned} \tag{23}$$

[3] This observation is denoted as *short memory principle* in [23].

respectively, where the inequalities

$$v_j(t) \leq x_j(t) \leq w_j(t) \tag{24}$$

hold for all $j \in \{1, \ldots, n\}$.

If the integration of a cooperative FDE model is reinitialized at a point $t = T > 0$, an observer-based approach

$${}_T\mathscr{D}_t^{(\nu)}\mathbf{z}(t) = \begin{bmatrix} \mathbf{f}_v\left(\tilde{\mathbf{v}}(t)\right) + \boldsymbol{\mu}_v(t) \\ \mathbf{f}_w\left(\tilde{\mathbf{w}}(t)\right) + \boldsymbol{\mu}_w(t) \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \mathbf{H} \cdot \begin{bmatrix} \mathbf{v}(t) - \tilde{\mathbf{v}}(t) \\ \mathbf{w}(t) - \tilde{\mathbf{w}}(t) \end{bmatrix} \tag{25}$$

with the augmented state vector

$$\mathbf{z}(t) = \begin{bmatrix} \tilde{\mathbf{v}}(t) \\ \tilde{\mathbf{w}}(t) \\ \boldsymbol{\mu}_v(t) \\ \boldsymbol{\mu}_w(t) \end{bmatrix} \in \mathbb{R}^{4n} \tag{26}$$

can be used to enhance the pseudo state enclosures and the truncation error bounds in comparison with the ones obtained by the integrator resetting approach according to the previous subsection that only employs temporally constant truncation error bounds.

The observer (25) is initialized with the pseudo state vector

$$\mathbf{z}(T) = \begin{bmatrix} \mathbf{v}(T) \\ \mathbf{w}(T) \\ -\boldsymbol{\mu}_T \\ \boldsymbol{\mu}_T \end{bmatrix}, \tag{27}$$

where the truncation error bounds $[-\boldsymbol{\mu}_T\,;\,\boldsymbol{\mu}_T]$ are computed as described in the previous subsection. Due to the inclusion of the truncation error bounds $\boldsymbol{\mu}_v(t)$ and $\boldsymbol{\mu}_w(t)$ by means of so-called integrator disturbance models in (25), leading to constant values if $\mathbf{H} = \mathbf{0}$, the case of the error quantification according to the previous subsection is included as a special case in this formulation. If $\mathbf{H} \neq \mathbf{0}$, the bounds $\boldsymbol{\mu}_v(t)$ and $\boldsymbol{\mu}_w(t)$ are enhanced in such a way that the *virtual measurements* and the enhanced bounds for the pseudo state variables approach each other as close as possible. For that purpose, the augmented system model (25) must be a valid interval observer.

To make the augmented system model (25) with the estimated lower and upper bounding trajectories $\tilde{\mathbf{v}}(t)$ and $\tilde{\mathbf{w}}(t)$ a valid observer, the gain matrix $\mathbf{H}$ needs to be chosen so that the error dynamics associated with the bounding trajectories remain asymptotically stable with

$$\left\|\mathbf{v}(t) - \tilde{\mathbf{v}}(t)\right\| \to 0 \quad \text{and} \quad \left\|\mathbf{w}(t) - \tilde{\mathbf{w}}(t)\right\| \to 0 \quad \text{for} \quad t \to \infty \tag{28}$$

and that

$$\left[\mathbf{v}(t)\ ;\ \mathbf{w}(t)\right] \subseteq \left[\tilde{\mathbf{v}}(t)\ ;\ \tilde{\mathbf{w}}(t)\right] \tag{29}$$

is ensured for all $t \geq T$.

For corresponding stability criteria for linear fractional differential equations, see the eigenvalue domains summarized in [11, 35]. Due to the fact that these domains can be expressed effectively by linear matrix inequality constraints and that nonlinear models can be bounded by quasi-linear system models with polytopic uncertainty representations, stability requirements for the gain matrix $\mathbf{H}$ cannot only be obtained for linear system models. They can also be obtained from the existing literature for nonlinear ones as it has been shown, for example in [3, 11], for the design of robust state estimation schemes for FDEs.

To ensure the enclosure property (29) and to verify the decoupled nature of the equations in (25) with respect to $\tilde{\mathbf{v}}(t)$ and $\tilde{\mathbf{w}}(t)$ as well as $\boldsymbol{\mu}_v(t)$ and $\boldsymbol{\mu}_w(t)$, respectively, the Jacobian of the right-hand side of the augmented model (25) with respect to the pseudo state vector $\mathbf{z}(t)$ needs to satisfy the sign property presented in the inequalities (22) with now $4n$ as the dimension of the augmented model.

3.3 Periodic Reset of Fractional Integrators and Their Application to Predictor–Corrector State Estimation

The observer approach from the previous subsection is the basis for a predictor–corrector technique for state estimation if measurements

$$\mathbf{y}(t_{\mathrm{m},k}) \in [\mathbf{y}](t_{\mathrm{m},k}) = \left[\underline{\mathbf{y}}(t_{\mathrm{m},k})\ ;\ \overline{\mathbf{y}}(t_{\mathrm{m},k})\right] \tag{30}$$

(including interval uncertainty to represent bounded measurement errors with unknown distributions) are available at the time instants $t_{\mathrm{m},k}$ in the form

$$\mathbf{g}\left(\mathbf{x}(t_{\mathrm{m},k})\right) \in \left[\underline{\mathbf{y}}(t_{\mathrm{m},k})\ ;\ \overline{\mathbf{y}}(t_{\mathrm{m},k})\right]. \tag{31}$$

Then, the same observer (25) as in the previous subsection is employed with *virtual measurements* obtained from a simulation of the original system dynamics. The *actual state measurements* $[\mathbf{y}](t_{\mathrm{m},k})$ are then used to tighten the bounds included in the pseudo state initialization $\mathbf{z}(T)$ at each point $T = t_{\mathrm{m},k}$.

This tightening is either obtained by a direct intersection of the measured intervals $[\mathbf{y}](t_{\mathrm{m},k})$ with the already computed state bounds $\left[\tilde{\mathbf{v}}(t_{\mathrm{m},k})\ ;\ \tilde{\mathbf{w}}(t_{\mathrm{m},k})\right]$ in the case of a direct pseudo state measurement or by applying a suitable contraction scheme [12] (forward–backward contractor or Krawczky-type contractor) to the relation (31), where the bounds for $\mathbf{x}(t_{\mathrm{m},k})$ are initialized with the interval $\left[\tilde{\mathbf{v}}(t_{\mathrm{m},k})\ ;\ \tilde{\mathbf{w}}(t_{\mathrm{m},k})\right]$ as in the first case. After this tightening step, the procedure is continued as described in the previous subsection, where the modification of the pseudo state reinitialization is the only modification in comparison with the previous subsection.

This approach allows for directly handling the continuous-time dynamics of the system model between two subsequent discrete time instants at which measured

data are available. In such a way, the sampling times both for the numerical evaluation of the FDE model and the measurements can be decoupled.

Remark 3. Cases in which the measurement step size is not an integer multiple of the numerical integration step, or in which the measurement times themselves are uncertain, can be handled with the same procedure as in [31].

Remark 4. Future work will aim at removing the precondition of cooperativity of the original as well as the observed system dynamics in (15) and (25). To solve this task, the so-called TNL approach for the parameterization of interval observers as derived in [38] is a promising solution which — due to its direct applicability to descriptor models — can extend the approach presented in this paper also to cases in which only some of the pseudo state variables are described by the explicit FDE models studied in (15) and others are expressed implicitly by using algebraic constraints. This approach is named after three matrices $\mathbf{T}$, $\mathbf{N}$, and $\mathbf{L}$ which are included in the observer as design degrees of freedom instead of using purely the observer gain for defining the observer dynamics.

4 Illustrating Example: Observer Approach

In this section, the observer-based identification of bounds for time-domain truncation errors of FDE models is presented for both a nonlinear academic simulation scenario and for a close-to-life quasi-linear model for the charging and discharging dynamics of Lithium-ion batteries.

4.1 Nonlinear Academic Benchmark System

4.1.1 Observer-Based Quantification of Time-Domain Truncation Errors

As a first example, consider the uncertain FDE model

$$x^{(0.5)}(t) = -x(t) - p \cdot x^3(t) \tag{32}$$

with the interval-based temporally constant initialization

$$x(t) \in [x_0] = [0.9\ ;\ 1.0] \quad \text{for} \quad t \leq 0 \tag{33}$$

and the uncertain, time-invariant parameter $p \in [0.1\ ;\ 0.2]$. This system model is simulated over the time interval $t \in [0\ ;\ 10]$ with integrator resets at the time instants

$$T \in \{T', 2T', 3T', \ldots\}\ , \quad \text{where} \quad T' = 1. \tag{34}$$

Due to its scalar nature, this system model satisfies the property of cooperativity, so that (without integrator resetting) the true pseudo state enclosure $x(t) \in \left[v(t)\ ;\ w(t)\right]$ according to (23) can be determined by means of the crips system models

$${}_0\mathscr{D}_t^{(0.5)} v(t) = -v(t) - \overline{p} \cdot v^3(t)\ , \quad v(t \leq 0) = \underline{x}_0 \tag{35}$$

and

$$ {}_0\mathscr{D}_t^{(0.5)} w(t) = -w(t) - \underline{p} \cdot w^3(t) \ , \quad w(t \leq 0) = \overline{x}_0 . \tag{36} $$

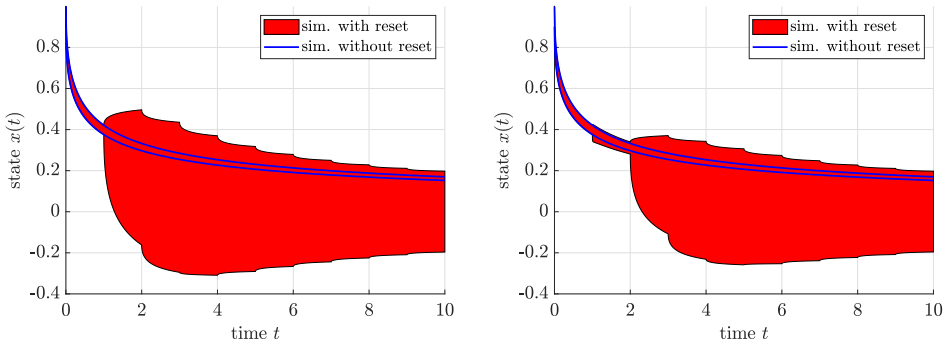
These bounds, computed with the help of the numerical integration routine `fde12` [7], are visualized by the solid lines in Figs. 2 and 3.

To investigate the observer approach given in Eq. (25), the time- and state-independent gain matrix

$$ \mathbf{H} = 20 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{37} $$

is chosen which ensures stability of the estimation error dynamics and cooperativity of the augmented pseudo state equations.

Fig. 2(a) presents a comparison between the integrator resetting in combination with piecewise constant bounds $\boldsymbol{\mu}_T$ for each time slice $t \in \left[(i-1)T' \ ; \ iT'\right]$, $i \in \{1, 2, \ldots, 10\}$ (determined according to (16), where $t_k = 0$ is chosen in each reinitialization point T defined in (34)), while Fig. 2(b) shows the observer-based enhancement of the pseudo state enclosures due to the temporal adaptation of the truncation error bounds according to Eq. (25) in Sec. 3.2.



(a) Piecewise constant error bounds $\boldsymbol{\mu}_T$. (b) Observer-based enhancement of $\boldsymbol{\mu}_v$, $\boldsymbol{\mu}_w$.

Figure 2: Simulation of the uncertain, nonlinear benchmark system (32).

4.1.2 Predictor–Corrector State Estimation

For the implementation of the predictor–corrector state estimator according to Sec. 3.3, we assume that pseudo state measurements are available at the time instants $t_{\mathrm{m},k} = T$ listed in (34).

The results in Fig. 3(a) distinguish the following two cases:

- The measured pseudo state information at the time instants $t_{\mathrm{m},k}$ corresponds to the enclosures from (35) and (36) with

$$y(t_{\mathrm{m},k}) \in \left[v(t_{\mathrm{m},k})\ ;\ w(t_{\mathrm{m},k})\right]. \tag{38}$$

  This scenario is depicted in Figs. 3(a) and 3(b).

- The measured pseudo state information at the time instants $t_{\mathrm{m},k}$ is obtained as

$$y(t_{\mathrm{m},k}) \in \hat{x}(t_{\mathrm{m},k}) + 0.001 \cdot [-1\ ;\ 1] \tag{39}$$

  with $\hat{x}(t)$ as the simulation of a nominal parameter model

$$\hat{x}^{(0.5)}(t) = -\hat{x}(t) - 0.15 \cdot \hat{x}^3(t) \tag{40}$$

  with $\hat{x}(t) = 0.95$ for $t \leq 0$, see Figs. 3(c) and 3(d).

From a comparison of Figs. 3(a) and 3(b), it is obvious that the observer-based approach in combination with resetting the pseudo state to the measured data leads to significantly tighter enclosures of the solutions than the use of piecewise constant error bounds $\boldsymbol{\mu}_T$.

A further tightening of the simulated bounds becomes possible if the uncertainty in the measured data is reduced in Figs. 3(c) and 3(d) in accordance with the second case above. Then, the new solution approach is capable of determining pseudo state enclosures that are partially tighter (in this case for the upper bounding trajectory) than a pure simulation of the uncertain nonlinear model (32) that still serves as the *virtual measurement* generator between the points T at which the *actual discrete-time measurements* are available. In such a way, the proposed observer-based enhancement of the time-domain truncation error bounds as well as the predictor–corrector state estimation scheme form the basis for the development of set-based parameter identification schemes that are part of our ongoing research activities.

4.2 Simplified Fractional Battery Model

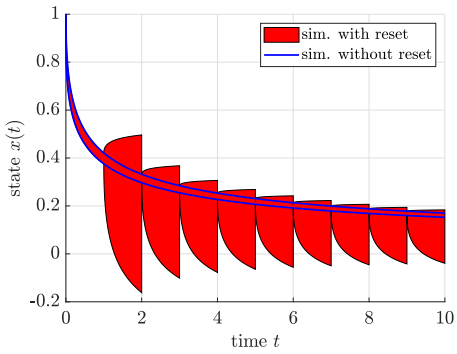
As a final application scenario, consider the fractional-order equivalent circuit model for the charging and discharging dynamics of Lithium-ion batteries depicted in Fig. 4.

Using the parameter values identified experimentally in [30], continuous-time state equations

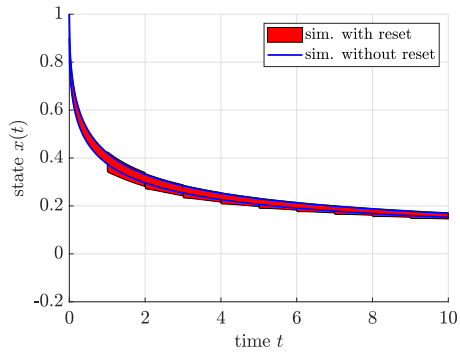
$${}_0\mathscr{D}_t^{(0.5)}\mathbf{x}(t) = \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{b} \cdot i(t) \tag{41}$$

with the system and input matrices

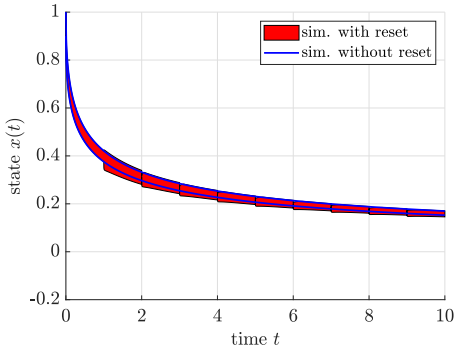
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\eta_1 \cdot \mathrm{sign}\left(i(t)\right)}{3600 C_{\mathrm{N}}} & 0 & 0 \\ 0 & 0 & -\frac{1}{RQ} \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 0 \\ -\frac{\eta_0}{3600 C_{\mathrm{N}}} \\ \frac{1}{Q} \end{bmatrix} \tag{42}$$



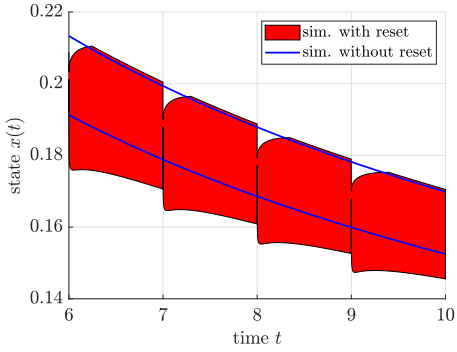
(a) Piecewise constant error bounds $\boldsymbol{\mu}_T$, measured pseudo state information according to Eq. (38).



(b) Observer-based enhancement of $\boldsymbol{\mu}_v$, $\boldsymbol{\mu}_w$ for the measured pseudo state information according to Eq. (38).



(c) Observer-based enhancement of $\boldsymbol{\mu}_v$, $\boldsymbol{\mu}_w$ for the measured pseudo state information according to Eq. (39).



(d) Enlarged view of Fig. 3(c).

Figure 3: Predictor–corrector state estimation for the uncertain, nonlinear benchmark system (32).

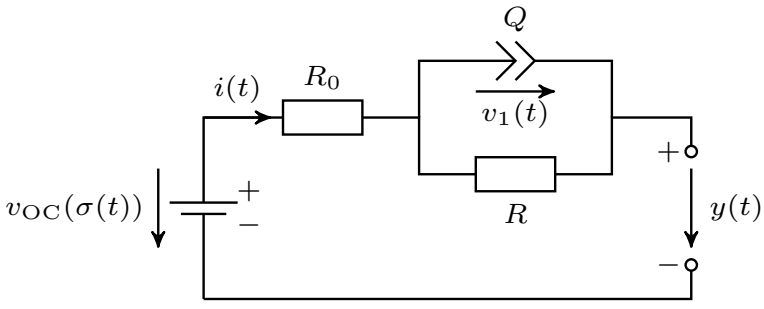


Figure 4: Equivalent circuit representation of a simplified fractional battery model.

as well as the pseudo state vector

$$\mathbf{x}(t) = \begin{bmatrix} \sigma(t) & {}_0\mathscr{D}_t^{(0.5)}\sigma(t) & v_1(t) \end{bmatrix}^T \in \mathbb{R}^3 \tag{43}$$

can be derived by applying Kirchhoff's voltage and current laws. In (43), $\sigma(t)$ denotes the state of charge of the battery, its fractional derivative is included in the vector $\mathbf{x}(t)$ to represent long-term memory phenomena, and $v_1(t)$ is the voltage across a non-integer constant phase element Q serving as a generalization of capacitors that are typically employed to represent polarization effects and the transportation of charge carriers in Thevenin equivalent circuit models of batteries [6].

For state estimation purposes, the terminal voltage (given in a quasi-linear representation)

$$y(t) = g\left(\mathbf{x}(t)\right) = \begin{bmatrix} \sum_{k=0}^{4} c_k \sigma^{k-1}(t) & 0 & -1 \end{bmatrix} \cdot \mathbf{x}(t) + \left(-R_0 + d_0 e^{d_1 \sigma(t)}\right) \cdot i(t) \tag{44}$$

is assumed to be available as a measured system output at specific discrete points in time.

To obtain further a cooperative system model, we consider the special case with $\eta_1 = 0$ and a controlled discharging process of the battery with the terminal current

$$i(t) = -\mathbf{k}^T \cdot \mathbf{x}(t) \tag{45}$$

as the system input in which the controller gain vector $\mathbf{k}^T$ is determined by pole assignment so that the eigenvalues of the closed-loop system are located at the points $\lambda \in \{0; -0.0002; -\frac{1}{RQ}\}$.

This leads to the linear autonomous system model

$${}_0\mathscr{D}_t^{(0.5)}\mathbf{x}(t) = \mathbf{A}_\mathrm{C} \cdot \mathbf{x}(t) \tag{46}$$

with

$$\mathbf{A}_\mathrm{C} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & a_{22} & 0 \\ 0 & a_{32} & a_{33} \end{bmatrix} \tag{47}$$

in which the entries a_{22}, a_{32}, and a_{33} are converted into interval parameters (displayed after outward rounding of the corresponding bounds) according to

$$\begin{aligned} a_{22} &\in [-0.000220\ ;\ -0.000179] \\ a_{32} &\in [\ \ 0.097557\ ;\ \ \ 0.119237] \\ a_{33} &\in [-0.531557\ ;\ -0.434910] \end{aligned} \tag{48}$$

to account for independent uncertainties of each of these quantities in the intervals of $\pm 10\%$ around the respective nominal values obtained with the help of the parameters given in [28] and [30].

The initial conditions of the system are assumed to be uncertain according to the Caputo definition

$$\mathbf{x}(t) = \begin{bmatrix} 0.5 & 0.01 & 0.1 \end{bmatrix}^T \cdot [0.9\ ;\ 1.1]\ , \quad t \leq 0. \tag{49}$$

To set up the observer-based enhancement of the time-domain truncation error bounds according to Eq. (25) with the resetting time instants

$$T \in \{T', 2T', 3T', \ldots\}\ , \quad \text{where} \quad T' = 60\,\text{s}, \tag{50}$$

the gain matrix

$$\mathbf{H} = 5 \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{51}$$

is chosen in the following subsections. According to the description in Sec. 3.2, this matrix is specified so that stability and cooperativity of the estimation error dynamics are ensured.

4.2.1 Observer-Based Quantification of Time-Domain Truncation Errors

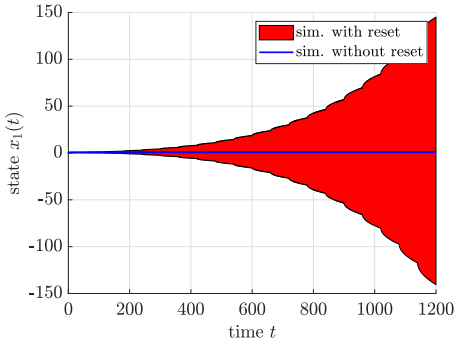
Fig. 5 summarizes a simulation of the fractional battery model in terms of a direct evaluation of the pseudo state equations (46)–(48) by directly exploiting the property of cooperativity. These results are shown by solid lines, indicating the lower and upper bounds of each state variable, respectively. As for the previous academic example, the numerical solver `fde12` [7] has been used for this purpose.

When resetting the fractional integrator at the time instants (50) and using constant bounds for the time-domain truncation errors, a rapid inflation of the pseudo state enclosures can be observed. This inflation is reduced in Fig. 6, where it has been assumed that the bounds, resulting from the cooperative system simulation, are available as initial conditions for each of the time slices. This resetting of the integrator, together with a reinitialization of the pseudo state for each point in time T is shown in the left column of Fig. 6 for piecewise constant bounds of the time-domain truncation errors.

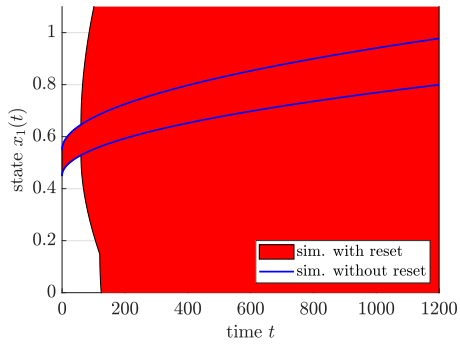
Activating the observer-based quantification of the truncation according to Sec. 3.2 additionally, as illustrated in the right column of Fig. 6, leads to significantly tighter outer enclosures that satisfy the relation (29) with certainty.

4.2.2 Predictor–Corrector State Estimation

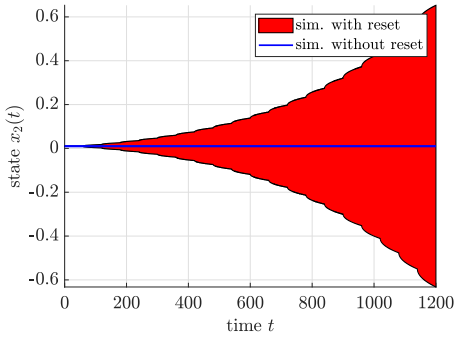
In practical situations, the resetting of the fractional integrator is often combined with a pseudo state estimation approach as presented in Sec. 3.3. To visualize the applicability of this technique for the model of a controlled Lithium-ion battery, it



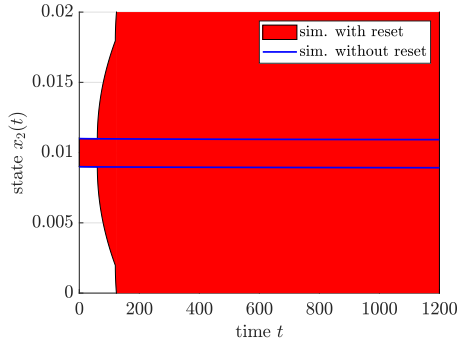
(a) State $x_1(t)$ (state of charge $\sigma(t)$).



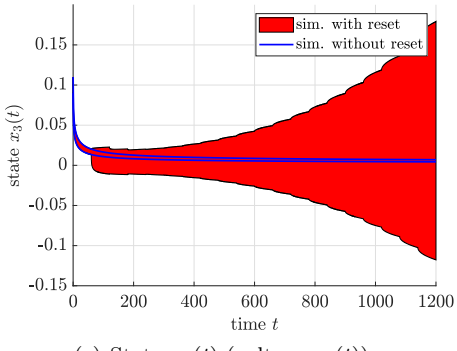
(b) State $x_1(t)$ (enlarged).



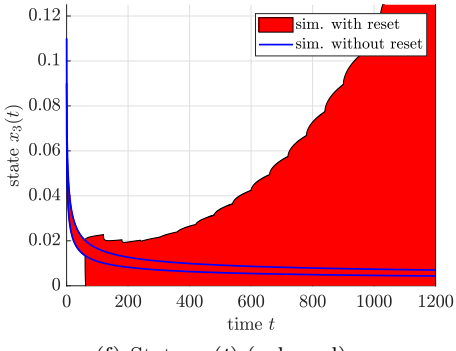
(c) State $x_2(t)$.



(d) State $x_2(t)$ (enlarged).

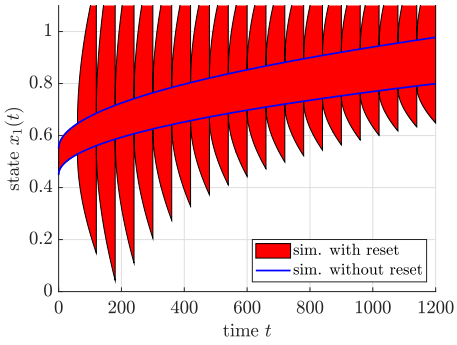


(e) State $x_3(t)$ (voltage $v_1(t)$).

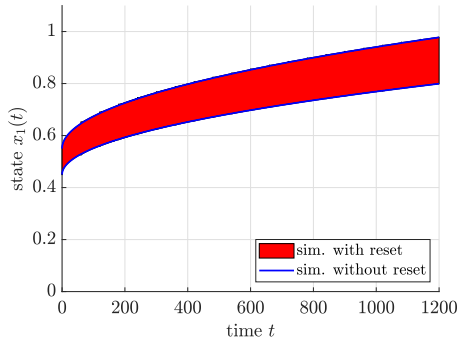


(f) State $x_3(t)$ (enlarged).

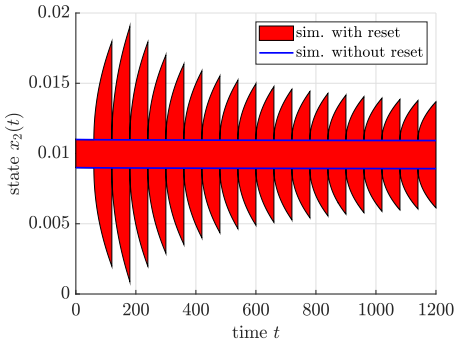
Figure 5: Simulation of the simplified battery model with piecewise constant error bounds $\boldsymbol{\mu}_T$; integrator reset at the points T defined in (50).



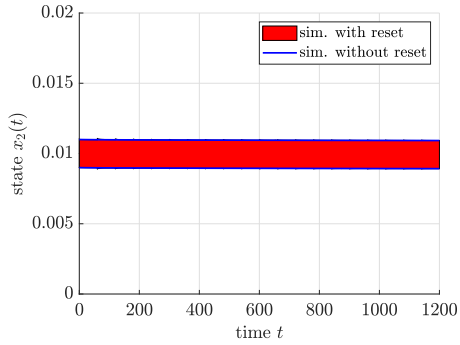
(a) State $x_1(t)$ (state of charge $\sigma(t)$.)



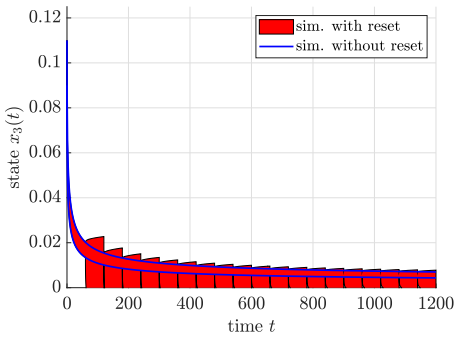
(b) State $x_1(t)$ (state of charge $\sigma(t)$).



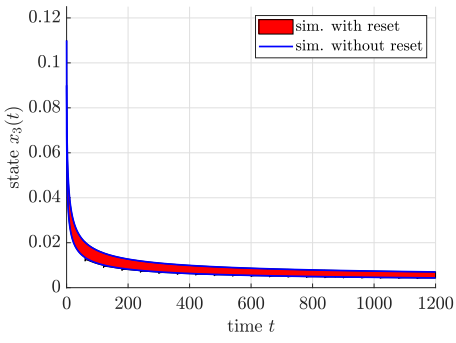
(c) State $x_2(t)$.



(d) State $x_2(t)$.



(e) State $x_3(t)$ (voltage $v_1(t)$).



(f) State $x_3(t)$ (voltage $v_1(t)$).

Figure 6: Comparison of piecewise constant error bounds $\boldsymbol{\mu}_T$ (left column) with the observer-based enhancement $\boldsymbol{\mu}_v$, $\boldsymbol{\mu}_w$ (right column) according to Sec. 3.2; resetting to the true state enclosures $\left[\mathbf{v}(t)\ ;\ \mathbf{w}(t)\right]$ at each time instant T defined in (50).

is assumed that uncertain measurements of the terminal voltage of the battery are available at each time instant $t_{\mathrm{m},k} = T$. These uncertain measurements are chosen as the intervals

$$[y](t_{\mathrm{m},k}) = \operatorname{mid}\left([g]\left(\left[\mathbf{v}(t_{\mathrm{m},k})\ ;\ \mathbf{w}(t_{\mathrm{m},k})\right]\right)\right) + [-10\ ;\ 10]\ \mathrm{mV}, \tag{52}$$

where $\operatorname{mid}\left([x]\right) = \frac{1}{2} \cdot (\underline{x} + \overline{x})$ defines the midpoint of an interval $[x]$.

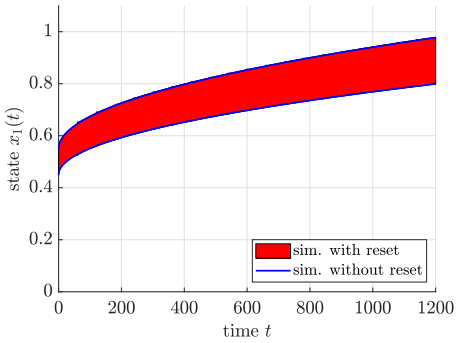
At each measurement instant T, the new pseudo state bounds are then initialized with the enclosure $\left[\tilde{\mathbf{v}}(T)\ ;\ \tilde{\mathbf{w}}(T)\right]$ obtained at the end of the previous time slice. From these bounds, an axis-aligned interval box is extracted by a SIVIA-like state reconstruction for the measured system output (44) that eliminates subboxes that are incompatible with the measurement intervals (52). To continue the simulation further, a tight axis-aligned interval hull around the not eliminated boxes is formed, so that the pseudo state enclosures shown in Fig. 7 are obtained.

Future work will make use of these bounds for an identification of interval parameters included in both the FDE model and the algebraic output equation of a dynamic system.

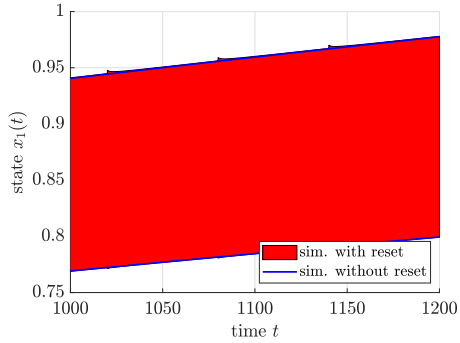
5 Conclusions

In this paper, a novel observer-based approach for the quantification of time-domain truncation errors of FDE models has been presented. These errors arise inevitably when resetting fractional integrators. Integrator resets are necessary for the numerical evaluation of FDE models both to restrict the growth of memory demands when evaluating FDEs over long time spans and to take into account measured state information at distinct points in time between which the system dynamics are evolving continuously.

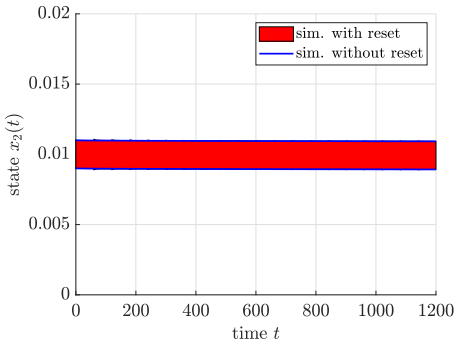
Future work will make use of the presented approach to solve the tasks of identifying past pseudo state information from an observed evolution of these quantities into the future and to identify uncertain system parameters on the basis of predictor–corrector state estimators. Moreover, the TNL interval observer design approach [38] for non-cooperative system models will be taken into consideration to avoid the currently existing necessity to transform non-cooperative models into cooperative ones by using the approaches presented in [14]. Although these approaches are useful for many practical applications, they always lead to conservative state enclosures due to the wrapping effect [17] that is inevitable when transforming the state equations and the domains of uncertain initial conditions with the help of (static) similarity transformations. This pessimism can be reduced by the TNL approach due to the introduction of further degrees of freedom for the observer parameterization. Moreover, this approach will also make the proposed methodology applicable to fractional descriptor systems.



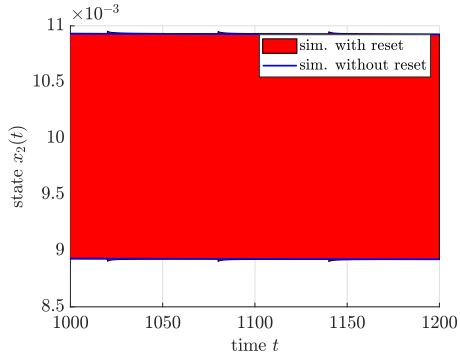
(a) State $x_1(t)$ (state of charge $\sigma(t)$.)



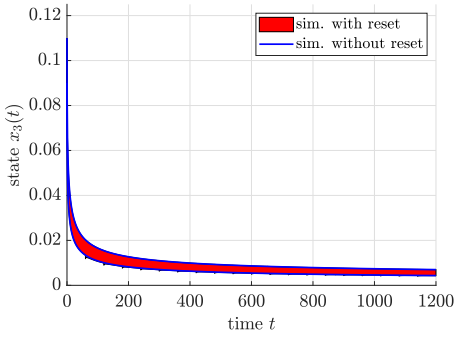
(b) State $x_1(t)$ (enlarged).



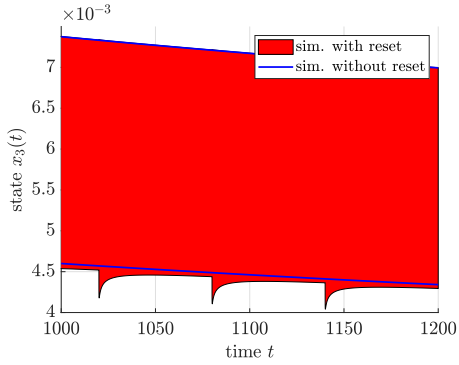
(c) State $x_2(t)$.



(d) State $x_2(t)$ (enlarged).



(e) State $x_3(t)$ (voltage $v_1(t)$).



(f) State $x_3(t)$ (enlarged).

Figure 7: Simulation using the observer-based enhancement $\boldsymbol{\mu}_v$, $\boldsymbol{\mu}_w$ according to Sec. 3.2 and contractor-based resetting of the state variables at each measurement instant T according to Sec. 3.3.

References

[1] Amairi, M., Aoun, M., Najar, S., and Abdelkrim, M.N. A constant enclosure method for validating existence and uniqueness of the solution of an initial value problem for a fractional differential equation. *Applied Mathematics and Computation*, 217(5):2162–2168, 2010. DOI: 10.1016/j.amc.2010.07.015.

[2] Andre, D., Meiler, M., Steiner, K., Wimmer, Ch., Soczka-Guth, T., and Sauer, D.U. Characterization of high-power lithium-ion batteries by electrochemical impedance spectroscopy. I. Experimental investigation. *Journal of Power Sources*, 196(12):5334–5341, 2011. DOI: 10.1016/j.jpowsour.2010.12.102.

[3] Bel Haj Frej, G., Malti, R., Aoun, M., and Raïssi, T. Fractional interval observers and initialization of fractional systems. *Communications in Nonlinear Science and Numerical Simulation*, 82:105030, 2020. DOI: 10.1016/j.cnsns.2019.105030.

[4] Delavari, H., Lanusse, P., and Sabatier, J. Fractional order controller design for a flexible link manipulator robot. *Asian Journal of Control*, 15(3):783–795, 2013. DOI: 10.1002/asjc.677.

[5] Efimov, D., Raïssi, T., Chebotarev, S., and Zolghadri, A. Interval state observer for nonlinear time varying systems. *Automatica*, 49(1):200–205, 2013. DOI: 10.1016/j.automatica.2012.07.004.

[6] Erdinc, O., Vural, B., and Uzunoglu, M. A dynamic lithium-ion battery model considering the effects of temperature and capacity fading. In *Proc. of International Conference on Clean Electrical Power*, pages 383–386, Capri, Italy, 2009. DOI: 10.1109/ICCEP.2009.5212025.

[7] Garrappa, R. Predictor-corrector PECE method for fractional differential equations. MATLAB Central File Exchange. URL: https://www.mathworks.com/matlabcentral/fileexchange/32918-predictor-corrector-pece-method-for-fractional-differential-equations (accessed: Feb. 22, 2022).

[8] Garrappa, R. Numerical evaluation of two and three parameter Mittag-Leffler functions. *SIAM Journal on Numerical Analysis*, 53(3):1350–1369, 2015. DOI: 10.1137/140971191.

[9] Goodwine, B. Modeling a multi-robot system with fractional-order differential equations. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1763–1768, Hong Kong, China, 2014. DOI: 10.1109/ICRA.2014.6907089.

[10] Haubold, H.J., Mathai, A.M., and Saxena, R.K. Mittag-Leffler functions and their applications. *Journal of Applied Mathematics*, 2011:298628, 2011. DOI: 10.1155/2011/298628.

[11] Hildebrandt, E., Kersten, J., Rauh, A., and Aschemann, H. Robust interval observer design for fractional-order models with applications to state estimation of batteries. In *Proc. of the 21st IFAC World Congress*, Berlin, Germany, 2020. DOI: 10.1016/j.ifacol.2020.12.2052.

[12] Jaulin, L., Kieffer, M., Didrit, O., and Walter, É. *Applied Interval Analysis*. Springer–Verlag, London, 2001. DOI: 10.1007/978-1-4471-0249-6.

[13] Kempfle, S., Schäfer, I., and Beyer, H. Fractional differential equations and viscoelastic damping. In *2001 European Control Conference (ECC)*, pages 1738–1743, 2001. DOI: 10.23919/ECC.2001.7076172.

[14] Kersten, J., Rauh, A., and Aschemann, H. State-space transformations of uncertain systems with purely real and conjugate-complex eigenvalues into a cooperative form. In *Proc. of 23rd Intl. Conference on Methods and Models in Automation and Robotics*, pages 797–802, Miedzyzdroje, Poland, 2018. DOI: 10.1109/MMAR.2018.8486085.

[15] Lanusse, P., Malti, R., and Melchior, P. CRONE control system design toolbox for the Control Engineering Community: Tutorial and case study. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 371:20120149, 2013. DOI: 10.1098/rsta.2012.0149.

[16] Lazarevic, M. P. Biologically inspired control and modeling of (Bio)Robotic systems and some applications of fractional calculus in mechanics. *Theoretical and Applied Mechanics*, 40:163–187, 2013. DOI: 10.2298/TAM1301163L.

[17] Lohner, R. On the ubiquity of the wrapping effect in the computation of the error bounds. In Kulisch, U., Lohner, R., and Facius, A., editors, *Perspectives on Enclosure Methods*, pages 201–217, Wien, New York, 2001. Springer–Verlag. DOI: 10.1007/978-3-7091-6282-8_12.

[18] Lyons, R., Vatsala, A.S., and Chiquet, R. Picard's iterative method for Caputo fractional differential equations with numerical results. *Mathematics*, 5(4), 2017. DOI: 10.3390/math5040065.

[19] Miyajima, S. Computing enclosures for the matrix Mittag–Leffler function. *Journal of Scientific Computing*, 87:62, 2021. DOI: 10.1007/s10915-021-01447-6.

[20] Monje, C.A., Vinagre, B.M., Feliu, V., and Chen, Y. Tuning and auto-tuning of fractional order controllers for industry applications. *Control Engineering Practice*, 16(7):798–812, 2008. DOI: 10.1016/j.conengprac.2007.08.006.

[21] Oustaloup, A. *La Dérivation Non Entière: Théorie, Synthèse et Applications*. Hermès, Paris, 1995. In French.

[22] Oustaloup, A., Mathieu, B., and Lanusse, P. The CRONE control of resonant plants: Application to a flexible transmission. *European Journal of Control*, 1(2):113–121, 1995. DOI: 10.1016/S0947-3580(95)70014-0.

[23] Podlubny, I. *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*. Mathematics in Science and Engineering. Academic Press, London, 1999.

[24] Podlubny, I. Fractional-order systems and $PI^{\lambda}D^{\mu}$-controllers. *IEEE Transactions on Automatic Control*, 44(1):208–214, 1999. DOI: 10.1109/9.739144.

[25] Raïssi, T., Efimov, D., and Zolghadri, A. Interval state estimation for a class of nonlinear systems. *IEEE Transactions on Automatic Control*, 57:260–265, 2012. DOI: 10.1109/TAC.2011.2164820.

[26] Rauh, A. and Jaulin, L. Novel techniques for a verified simulation of fractional-order differential equations. *Fractal and Fractional*, 5(1):17, 2021. DOI: 10.3390/fractalfract5010017.

[27] Rauh, A. and Kersten, J. Toward the development of iteration procedures for the interval-based simulation of fractional-order systems. *Acta Cybernetica*, 25(1):21–48, 2020. DOI: 10.14232/actacyb.285660.

[28] Rauh, A. and Kersten, J. Verification and reachability analysis of fractional-order differential equations using interval analysis. In Dang, Thao and Ratschan, Stefan, editors, *Proceedings 6th International Workshop on Symbolic-Numeric methods for Reasoning about CPS and IoT*, Volume 331 of *Electronic Proceedings in Theoretical Computer Science*, pages 18–32. Open Publishing Association, 2021. DOI: 10.4204/EPTCS.331.2.

[29] Rauh, A., Kersten, J., and Aschemann, H. Interval-based verification techniques for the analysis of uncertain fractional-order system models. In *Proc. of the 18th European Control Conference ECC2020*, St. Petersburg, Russia, 2020. DOI: 10.23919/ECC51009.2020.9143758.

[30] Reuter, J., Mank, E., Aschemann, H., and Rauh, A. Battery state observation and condition monitoring using online minimization. In *Proc. of 21st Intl. Conference on Methods and Models in Automation and Robotics*, pages 797–802, Miedzyzdroje, Poland, 2016. DOI: 10.1109/MMAR.2016.7575313.

[31] Rohou, S. and Jaulin, L. Exact bounded-error continuous-time linear state estimator. *Systems & Control Letters*, 153:104951, 2021. DOI: 10.1016/j.sysconle.2021.104951.

[32] Scherer, R., Kalla, S.L., Tang, Y., and Huang, J. The Grünwald–Letnikov method for fractional differential equations. *Computers & Mathematics with Applications*, 62(3):902–917, 2011. DOI: 10.1016/j.camwa.2011.03.054.

[33] Smith, H.L. *Monotone Dynamical Systems: An Introduction to the Theory of Competitive and Cooperative Systems*, Volume 41. Mathematical Surveys and Monographs, American Mathematical Soc., Providence, 1995. DOI: 10.1090/surv/041.

[34] Tavares, D., Almeida, R., and Torres, D.F.M. Caputo derivatives of fractional variable order: Numerical approximations. *Communications in Nonlinear Science and Numerical Simulation*, 35:69–87, 2016. DOI: 10.1016/j.cnsns.2015.10.027.

[35] Trigeassou, J.-C. and Maamri, N. *Analysis, Modeling and Stability of Fractional Order Differential Systems 2*. John Wiley & Sons, Ltd, Hoboke, NJ, USA, 2019. DOI: 10.1002/9781119686859.

[36] Troparevsky, M., Seminara, S., and Fabio, M. *A Review on Fractional Differential Equations and a Numerical Method to Solve Some Boundary Value Problems*. In *Nonlinear Systems — Theoretical Aspects and Recent Applications*. IntechOpen, 2020. DOI: 10.5772/intechopen.86273.

[37] Wang, B., Liu, Z., Li, S., Moura, S., and Peng, H. State-of-charge estimation for lithium-ion batteries based on a nonlinear fractional model. *IEEE Trans. on Control Systems Technology*, 25(1):3–11, 2017. DOI: 10.1109/TCST.2016.2557221.

[38] Wang, Z., Lim, C.-C., and Shen, Y. Interval observer design for uncertain discrete-time linear systems. *Systems & Control Letters*, 116:41–46, 2018. DOI: 10.1016/j.sysconle.2018.04.003.

[39] Yameni Noupoue, Y.Y., Tandoğdu, Y., and Awadalla, M. On numerical techniques for solving the fractional logistic differential equation. *Advances in Difference Equations*, 2019, 2019. DOI: 10.1186/s13662-019-2055-y.

[40] Zou, Ch., Zhang, L., Hu, X., Wang, Z., Wik, T., and Pecht, M. A review of fractional-order techniques applied to lithium-ion batteries, lead-acid batteries, and supercapacitors. *Journal of Power Sources*, 390:286–296, 2018. DOI: 10.1016/j.jpowsour.2018.04.033.

Acta Cybernetica **26** (2023) 129–147.

Affine Iterations and Wrapping Effect: Various Approaches

Nathalie Revol[a]

Abstract

Affine iterations of the form $x_{n+1} = Ax_n + b$ converge, using real arithmetic, if the spectral radius of the matrix A is less than 1. However, substituting interval arithmetic to real arithmetic may lead to divergence of these iterations, in particular if the spectral radius of the absolute value of A is greater than 1. We will review different approaches to limit the overestimation of the iterates, when the components of the initial vector x_0 and b are intervals. We will compare, both theoretically and experimentally, the widths of the iterates computed by these different methods: the naive iteration, methods based on the QR- and SVD-factorization of A, and Lohner's QR-factorization method. The method based on the SVD-factorization is computationally less demanding and gives good results when the matrix is poorly scaled, it is superseded either by the naive iteration or by Lohner's method otherwise.

Keywords: interval analysis, affine iterations, matrix powers, Lohner's QR algorithm, QR factorization, SVD factorization

1 Introduction

The problem we consider is the evaluation of the successive iterates of

$$\begin{cases} x_{n+1} & = & Ax_n + b, \\ x_0 \text{ given}, & & \end{cases}$$

where $A \in \mathbb{R}^{d\times d}$, $x_n \in \mathbb{R}^d$ for every $n \in \mathbb{N}$ and $b \in \mathbb{R}^d$. More specifically, the focus is on the use of interval arithmetic to evaluate these iterates.

In what follows, interval quantities will be denoted in boldface.

[a]INRIA – LIP laboratory (Unité Mixte de Recherche 5668 Centre National de la Recherche Scientifique – École Normale Supérieure de Lyon – INRIA – Université Claude Bernard de Lyon), École Normale Supérieure de Lyon, France, E-mail: Nathalie.Revol@inria.fr, ORCID: 0000-0002-2503-2274

DOI: 10.14232/actacyb.295251

1.1 A Toy Example

This problem was brought to us through this example of an IIR (Infinite Impulse Response) linear filter in a state-space form:

$$x_n = 1.8 * x_{n-1} - 0.9 * x_{n-2} + 4.7.10^{-2} * (u_{n-2} + u_{n-1} + u_n)$$

for $x_0 = 0$ and $x_1 \in [1, 1.1]$. We assume that $u_n \in \mathbf{u} = [9.95\,,\,10.05]$ for every n.

This iteration can also be written as a linear recurrence in $\mathbb{R}^2$:

$$\begin{pmatrix} x_{n-1} \\ x_n \end{pmatrix} = A. \begin{pmatrix} x_{n-2} \\ x_{n-1} \end{pmatrix} + b_n,$$

where $A = \begin{pmatrix} 0 & 1 \\ -0.9 & 1.8 \end{pmatrix}$ and $b_n = \begin{pmatrix} 0 \\ 4.7.10^{-2} * (u_{n-2} + u_{n-1} + u_n) \end{pmatrix}$.

This toy example will be used to illustrate the various approaches mentioned in this paper. The first iterates, obtained using floating-point arithmetic, with random values for $x_1 \in [1, 1.1]$ and each $u_n \in [9.95\,,\,10.05]$, are given on the left two columns of Table 1.

The system stabilizes around 14, with variations due to the random values taken by the u_n. However, the following snippet of Octave code computes the successive iterates using interval arithmetic, using the interval $[1\,,\,1.1]$ for $\mathbf{x}_1$ and $\mathbf{u} = [9.95\,,\,10.05]$ for the u_n, that is, we replace $u_{n-2} + u_{n-1} + u_n$ by $3 * \mathbf{u}$.

```
A=[[0 1];[-0.9 1.8]];
xn=[infsup(0,0);infsup(1,1.1)];
b=4.7e-2 * 3.0*[infsup(0,0);infsup(9.95,10.05)];
n=500; for i=1:n, i , xn=A*xn+b, wid(xn(1)), end;
```

Table 1: Comparison of the behavior of the iterates: point values on the left, interval values on the right

n	x_n	n	x_n	n	$\mathrm{wid}(\mathbf{x}_n)$	n	$\mathrm{wid}(\mathbf{x}_n)$
0	0	20	9.1518	0	0	20	$3.2293.10^{5}$
1	1.0617	30	17.0186	1	0.1000	30	$8.8808.10^{8}$
2	3.3183	40	12.4414	2	0.1941	40	$2.4423.10^{12}$
3	6.4234	50	15.0305	3	0.4535	50	$6.7164.10^{15}$
4	9.9851	60	13.6130	4	1.0051	60	$1.8470.10^{19}$
5	13.6031	70	14.3858	5	2.2313	70	$5.0794.10^{22}$
6	16.9117	80	13.9680	6	4.9350	80	$1.3969.10^{26}$
7	19.6103	90	14.1510	7	10.905	90	$3.8415.10^{29}$
8	21.4884	100	14.0949	8	24.085	100	$1.0564.10^{33}$
9	22.4394	200	14.0870	9	53.182	200	$2.6137.10^{67}$
10	22.4595	300	14.1443	10	117.42	300	$6.4663.10^{101}$
12	20.1508	400	14.1282	12	572.31	400	$1.5998.10^{136}$
15	13.8931	500	14.0828	15	6158.0	500	$3.9580.10^{170}$

On the right two columns of Table 1, the widths of the successive iterates $\mathbf{x}_n$ are given: the widths of the iterates diverge rapidly to infinity.

The explanation of this phenomenon is the following: the spectral radius of A is strictly less than 1: $\rho(A) \simeq 0.9487 < 1$, and thus the exact (and, for that matter, floating-point) iterations converge. However, the recurrence satisfied by the widths of the iterates is $\mathrm{wid}(\mathbf{x}_n) = 1.8 * \mathrm{wid}(\mathbf{x}_{n-1}) + 0.9 * \mathrm{wid}(\mathbf{x}_{n-2}) + 4.7.10^{-2} * 3 * \mathrm{wid}(\mathbf{u})$, which corresponds to the 2-dimensional iteration $w_n = |A|.w_{n-1} + w_b$, with $w_n = \mathrm{wid}(\mathbf{x}_n)$, $|A|$ the matrix whose coefficients are the absolute values of the coefficients of A and $w_b = 4.7.10^{-2} * 3 * \mathrm{wid}(\mathbf{u})$. As the spectral radius of $|A|$ is larger than 1, indeed $\rho(|A|) \simeq 2.208 > 1$, the iterations diverge.

This phenomenon is a special case of the so-called *wrapping effect*. Its ubiquity in interval computations has been put in evidence by Lohner in [8].

1.2 The Wrapping Effect

The wrapping effect is ubiquitous, as defined and developed in [8]. It can be described as the overestimation due to the enclosure of the sought set in a set of a given simple structure. In our case, this simple structure corresponds to multidimensional intervals or boxes, that is, parallelepipeds with sides parallel to the axes of the coordinate system. When the computation is iterative, and when each iteration produces such an overestimating set that is used as the starting point of the next iteration, the size of the computed set may grow exponentially in the number of iterations, even when the exact solution set remains bounded and small.

Lohner also put in evidence that the affine iteration we study in this paper, namely $x_{n+1} = Ax_n + b$, or more generally $x_{n+1} = A_n x_n + b_n$ with x_{n+1}, x_n and b_n vectors in $\mathbb{R}^d$ and $A_n \in \mathbb{R}^{d \times d}$ for every $n \in \mathbb{N}$, is archetypal. It occurs in many algorithms, and the examples cited in [8] include

- matrix-vector iterations as the ones studied in this paper;
- discrete dynamical systems: $\mathbf{x}_{n+1} = f(\mathbf{x}_n)$, $\mathbf{x}_0$ given and f sufficiently smooth;
- continuous dynamical systems (ODEs): $x'(t) = g(t, x(t))$, $x(0) = x_0$, which is studied through a numerical one step method (or more) of the kind $\mathbf{x}_{n+1} = \mathbf{x}_n + h\Phi(\mathbf{x}_n, t_n) + \mathbf{z}_{n+1}$;
- difference equations: $a_0\mathbf{z}_n + a_1\mathbf{z}_{n+1} + \ldots + a_m\mathbf{z}_{n+m} = b_n$ with $\mathbf{z}_1, \ldots \mathbf{z}_m$ given;
- linear systems with (banded) triangular matrix;
- automatic differentiation.

In this paper, we concentrate on examples similar to the toy example presented above: for every initial value $x_0 \in \mathbb{R}^d$, the sequence of iterates $(x_n)_{n \in \mathbb{N}}$ converges to a finite value $x^* \in \mathbb{R}^d$, since $\rho(A) < 1$; however, the computations performed using interval arithmetic diverge because their behaviour is dictated by $\rho(|A|)$ which is larger than 1. We are interested in the iterates computed using interval arithmetic:

it is established that these iterates increase in width, however different approaches can be applied to counteract the exponential growth of the width of the iterates. Several of them, some new as in Sections 2.3.2 and 2.3.3, and some already well established as in Section 2.4, will be tried and compared, in terms of the widths of the results and the computational time.

2 Theoretical Results

2.1 Problem and Notations

Let A be a $d \times d$ matrix in $\mathbb{R}^{d\times d}$, $\mathbf{x}_0 \in \mathbb{IR}^d$ be an interval vector (boldface font is used for interval quantities and $\mathbb{IR}$ stands for the set of real intervals), x_0 a vector in $\mathbb{R}^d$ with $x_0 \in \mathbf{x}_0$, $\mathbf{b} \in \mathbb{IR}^d$ an interval vector, and b a vector in $\mathbb{R}^d$ and $b \in \mathbf{b}$. In what follows, n denotes the number of iterations.

It is assumed that $\rho(A) < 1$ and $\rho(|A|) > 1$. The fact that $\rho(A) < 1$ implies that $A^n \to 0$ when $n \to +\infty$.

A first goal is to determine the set of all fixed-points of the iteration

$$\begin{cases} x_0 \in \mathbf{x}_0,\ b \in \mathbf{b}, \\ x_{n+1} = Ax_n + b \end{cases}$$

for every $x_0 \in \mathbf{x}_0$ and every $b \in \mathbf{b}$.

It is known that x_n can be written as

$$x_n = A^n x_0 + \sum_{i=1}^{n-1} A^i b,$$

thus

$$\{x_n : x_0 \in \mathbf{x}_0\} \subset A^n \mathbf{x}_0 + \left(\sum_{i=1}^{n-1} A^i\right) \mathbf{b}.$$

However, when the vectors x_0 and b are replaced in the iterative formula by their interval enclosures $\mathbf{x}_0$ and $\mathbf{b}$, one obtains the new interval vector $\mathbf{x}_{n+1}$, which is computed as:

$$\begin{cases} \mathbf{x}_0 \text{ and } \mathbf{b} \text{ given}, \\ \mathbf{x}_{n+1} = A\mathbf{x}_n + \mathbf{b}. \end{cases}$$

Another goal is to determine a tight enclosure for each iterate of this diverging set of intervals.

As mentioned above, the increase in widths of the iterates can be attributed to the use of parallelepipeds with sides parallel to the axes of the coordinate system, and not to the geometry of the transformation. To cure this problem, changes of coordinates will be applied, using an invertible matrix B, with $x = By \Leftrightarrow y = B^{-1}x$ and its interval counterpart $\mathbf{x} = B\mathbf{y}$. This yields the iteration

$$\begin{cases} x_{n+1} &= By_{n+1} \\ y_{n+1} &= B^{-1}ABy_n + B^{-1}b \end{cases}$$

and its interval counterpart

$$\begin{cases} \mathbf{x}_{n+1} &= B\mathbf{y}_{n+1} \\ \mathbf{y}_{n+1} &= B^{-1}AB\mathbf{y}_n + B^{-1}\mathbf{b}. \end{cases}$$

In what follows, analogously to the approach in [10], to establish bounds and to simplify the derivation of their proofs, we assume A diagonalizable (this will not necessarily be the case for the experiments) and A can be diagonalized as $A = P^{-1}\Lambda P$ where Λ is a diagonal matrix with the eigenvalues $\lambda_1, \ldots \lambda_d$ of A on the diagonal and the columns of P^{-1} are the corresponding eigenvectors.

The iteration considered in this paper corresponds to $x_n = A^n x_0 + \sum_{i=0}^{n-1} A^i b$. The numerical unstability of computing the matrix power A^n and applying it to a vector, is well known: $A^n x_0$ tends to be aligned with the eigenvector of A associated with the largest (in module) eigenvalue, and the information corresponding to the contribution of the other eigenvectors is lost. To avoid this well-known problem of the power method, we will consider orthogonal changes of coordinates. The choice of the orthogonal matrices is related to A, the matrix of the iteration.

We will first consider the QR-factorization of A: $A = QR$ with $Q \in \mathbb{R}^{d\times d}$ orthogonal, that is, $QQ' = Q'Q = I$ is the identity matrix and $R \in \mathbb{R}^{d\times d}$ is upper triangular.

The other factorization used in this paper is the SVD-factorization of A: $A = U\Sigma V'$ with U, V and $\Sigma \in \mathbb{R}^{d\times d}$, where U and V are orthogonal and Σ is a diagonal matrix with the singular values $\sigma_1, \ldots \sigma_d$ of A on the diagonal. We also assume that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$. This idea has been sketched but not completely developed by Beaumont in [2].

2.2 Known results

Mayer and his co-authors have extensively studied the existence of a fixed-point for the iteration studied in this paper. In [9], Mayer and Warnke have thoroughly established formulas for the fixed-point in the case of $\rho(|A|) < 1$: this fixed-point is independent of the starting interval $\mathbf{x}_0$. In [1], Arndt and Mayer have established necessary and sufficient condition on A for a fixed-point to exist, when $\rho(|A|) = 1$. In this case, the fixed-point is an interval of nonzero width, that is, a non-degenerate interval. It is well-known that the widths of the iterates diverge when $\rho(|A|) > 1$, and thus that no fixed-point exists in this case. Our goal is to study the speed of divergence of the iterates in this case.

2.3 Different Approaches along with Theoretical Bounds

The main idea is to use an orthogonal change of coordinates which is related to the matrix of the iteration. As the matrix A is kept constant for all iterations (and this is not the case in the more general approach of Lohner, see Section 2.4), the change of coordinates is also kept constant and given by an orthogonal matrix B.

The two orthogonal matrices considered in what follows are either $B = Q$ from the QR-factorization of A, or $B = U$, resp. $B = V'$, from the SVD-factorization of A.

2.3.1 Orthogonal Change of Coordinates

Before diving into the specificities of these changes of coordinates, let us study the general change of coordinates using an orthogonal matrix B, that is, $B^{-1} = B'$, with $x = By \Leftrightarrow y = B^{-1}x$ and its interval counterpart $\mathbf{x} = B\mathbf{y}$. The interval iteration is

$$\left\{ \begin{array}{lcl} \mathbf{x}_{n+1} & = & B\mathbf{y}_{n+1} \\ \mathbf{y}_{n+1} & = & B^{-1}AB\mathbf{y}_n + B^{-1}\mathbf{b}. \end{array} \right.$$

Thus the iteration satisfied by the width of the $\mathbf{y}_n$ is

$$\begin{array}{rcl} \mathrm{wid}(\mathbf{y}_{n+1}) & = & |B^{-1}AB|\mathrm{wid}(\mathbf{y}_n) + |B^{-1}|\mathrm{wid}(\mathbf{b}) \\ & \leq & |B^{-1}|.|A|.|B|\mathrm{wid}(\mathbf{y}_n) + |B^{-1}|\mathrm{wid}(\mathbf{b}) \end{array}$$

where the inequalities are to be understood componentwise. By induction on n,

$$\mathrm{wid}(\mathbf{y}_n) \leq (|B^{-1}|.|A|.|B|)^n.\mathrm{wid}(\mathbf{y}_0) + \sum_{i=0}^{n-1}(|B^{-1}|.|A|.|B|)^i.|B^{-1}|.\mathrm{wid}(\mathbf{b}).$$

Taking norms, one gets

$$\begin{array}{rcl} \|\mathrm{wid}(\mathbf{y}_n)\| & \leq & \left(\| \, |B^{-1}| \, \|.\| \, |A| \, \|.\| \, |B| \, \|\right)^n \|.\mathrm{wid}(\mathbf{y}_0)\| \\ & + & \sum_{i=0}^{n-1} \left(\| \, |B^{-1}| \, \|.\| \, |A| \, \|.\| \, |B| \, \|\right)^i .\| \, |B^{-1}| \, \|.\|\mathrm{wid}(\mathbf{b})\| \\ & \leq & \left(\| \, |B^{-1}| \, \|.\| \, |A| \, \|.\| \, |B| \, \|\right)^n \|.\mathrm{wid}(\mathbf{y}_0)\| \\ & + & \frac{\left(\| \, |B^{-1}| \, \|.\| \, |A| \, \|.\| \, |B| \, \|\right)^n - 1}{\| \, |B^{-1}| \, \|.\| \, |A| \, \|.\| \, |B| \, \| - 1} \| \, |B^{-1}| \, \|.\|\mathrm{wid}(\mathbf{b})\|. \end{array}$$

Remark: if the considered norm is the matrix norm induced by the vector Euclidean norm, then $\| \, |B| \, \|_2 = \|B\|_2$ for any matrix B. Similarly, $\| \, |B| \, \|_\infty = \|B\|_\infty \leq \sqrt{d}$ for any $d \times d$ orthogonal matrix B. In such cases, the bound becomes

$$\begin{array}{rcl} \|\mathrm{wid}(\mathbf{y}_n)\| & \leq & (\kappa(B).\| \, |A| \, \|)^n \, \|\mathrm{wid}(\mathbf{y}_0)\| \\ & + & \frac{(\kappa(B).\| \, |A| \, \|)^n - 1}{\kappa(B).\| \, |A| \, \| - 1} \|B^{-1}\|.\|\mathrm{wid}(\mathbf{b})\|, \end{array}$$

where $\kappa(B)$ denotes $\|B\|.\|B^{-1}\|$, the condition number of B for the problem of solving a linear system.

Since $\|B\|_2 = \|B^{-1}\|_2 = \kappa_2(B) = 1$ for an orthogonal matrix B, this bound simplifies even further with the Euclidean norm:

$$\|\mathrm{wid}(\mathbf{y}_n)\| \leq \|A\|^n.\|\mathrm{wid}(\mathbf{y}_0)\| + \frac{\|A\|^n - 1}{\|A\| - 1}.\|\mathrm{wid}(\mathbf{b})\|.$$

In other words, theoretically there is no difference in the bounds on the widths of the iterates, whether an orthogonal change of coordinates takes place or not.

In what follows, we assume again A diagonalizable so as to simplify the presented proofs (this will not necessarily be the case for the experiments) and A can be diagonalized as $A = P^{-1}\Lambda P$ where Λ is diagonal. If we replace A by $P^{-1}\Lambda P$ in the iteration, one gets the mathematically equivalent formulation

$$\begin{array}{rcl} \mathbf{y}_{n+1} & = & B^{-1}P^{-1}\Lambda PB\mathbf{y}_n + B^{-1}\mathbf{b} \\ & = & (PB)^{-1}\Lambda(PB)\mathbf{y}_n + B^{-1}\mathbf{b}, \end{array}$$

thus

$$\mathrm{wid}(\mathbf{y}_n) = (|(PB)^{-1}|.|\Lambda|.|PB|).\mathrm{wid}(\mathbf{y}_n) + |B^{-1}|.\mathrm{wid}(\mathbf{b}),$$

and by induction

$$\mathrm{wid}(\mathbf{y}_n) = (|(PB)^{-1}|.|\Lambda|.|PB|)^n.\mathrm{wid}(\mathbf{y}_0) + \sum_{i=0}^{n-1}(|(PB)^{-1}|.|\Lambda|.|PB|)^i.|B^{-1}|.\mathrm{wid}(\mathbf{b}).$$

Taking the Euclidean norm of vectors and the induced matrix norm, one gets

$$\begin{array}{rcl} \|\mathrm{wid}(\mathbf{y}_n)\|_2 & \leq & (\kappa_2(PB)\|\Lambda\|_2)^n.\|\mathrm{wid}(\mathbf{y}_0)\|_2 \\ & + & \frac{(\kappa_2(PB)\|\Lambda\|_2)^n-1}{\kappa_2(PB)\|\Lambda\|_2-1}.\|\mathrm{wid}(\mathbf{b})\|_2. \end{array}$$

Let us note that $\kappa(PB) = \kappa(P)$. Furthermore, as Λ is diagonal, $\|\Lambda\|$ is the largest eigenvalue (in module) of A, that is, $\|\Lambda\| = \rho(A) < 1$. This implies

$$\|\mathrm{wid}(\mathbf{y}_n)\|_2 \leq (\kappa_2(P)\rho(A))^n.\|\mathrm{wid}(\mathbf{y}_0)\|_2 + \frac{(\kappa_2(P)\rho(A))^n-1}{\kappa_2(P)\rho(A)-1}.\|\mathrm{wid}(\mathbf{b})\|_2,$$

This inequality puts in evidence the influence of the condition number of P, the matrix of eigenvectors. For instance, in the ideal case where the eigenvectors form an orthonormal basis, no overestimation occurs.

2.3.2 Use of the QR Factorization

When the orthogonal change of coordinates involves Q from the QR-factorization of A, the algorithm can be written as

$$\begin{array}{rcl} A & = & QR, \\ x_{n+1} & = & Qy_{n+1} \\ \Leftrightarrow y_{n+1} & = & Q'x_{n+1} \\ y_{n+1} & = & Q'AQy_n + Q'b \end{array}$$

In exact arithmetic, one should get

$$y_{n+1} = RQy_n + Q'b.$$

The interval counterpart is

$$\begin{array}{rcl} \mathbf{x}_{n+1} & = & Q\mathbf{y}_{n+1} \\ \mathbf{y}_{n+1} & = & Q'AQ\mathbf{y}_n + Q'\mathbf{b}. \end{array}$$

2.3.3 Use of the SVD Factorization

Our second and third proposals consist in using respectively U and V from the SVD-factorization of A: from $A = U\Sigma V'$, we use either $B = U$ or $B = V'$, which yields

$$\begin{array}{rcl} x_{n+1} & = & Uy_{n+1} \\ \Leftrightarrow y_{n+1} & = & U'x_{n+1}, \\ y_{n+1} & = & U'AUy_n + U'b. \end{array}$$

In exact arithmetic, this corresponds to

$$y_{n+1} = \Sigma VUy_n + U'b.$$

The interval counterpart is

$$\begin{array}{rcl} \mathbf{x}_{n+1} & = & U\mathbf{y}_{n+1} \\ \mathbf{y}_{n+1} & = & U'AU\mathbf{y}_n + U'\mathbf{b}. \end{array}$$

$$\begin{array}{rcl} x_{n+1} & = & V'y_{n+1} \\ \Leftrightarrow y_{n+1} & = & Vx_{n+1}, \\ y_{n+1} & = & VAV'y_n + Vb. \end{array}$$

In exact arithmetic, this corresponds to

$$y_{n+1} = VU\Sigma y_n + Vb.$$

The interval counterpart is

$$\begin{array}{rcl} \mathbf{x}_{n+1} & = & V'\mathbf{y}_{n+1} \\ \mathbf{y}_{n+1} & = & VAV'\mathbf{y}_n + V\mathbf{b}. \end{array}$$

Remark: VU is also an orthogonal matrix.

2.4 Lohner's QR Method

A well-known approach is given in Lohner, e.g. in [8] and studied in details by Nedialkov and Jackson in [10]. It is usually presented for the iteration $x_{n+1} = A_n x_n + b_n$, that is when the matrix and the affine term vary at each iteration.

Lohner's QR method consists in performing the following iteration:

$$\left\{ \begin{array}{ll} y_0 = x_0,\ Q_0 = I,\ [Q_1, R_1] = qr(A) \text{ that is, } A = Q_1R_1 \\ [Q_{n+1}, R_{n+1}] & = qr(R_nQ_n) \\ y_{n+1} & = Q'_{n+1}AQ_ny_n + Q'_{n+1}b \\ x_{n+1} & = Q_{n+1}y_{n+1} \end{array} \right.$$

and its interval counterpart is

$$\left\{ \begin{array}{ll} \mathbf{y}_0 = \mathbf{x}_0,\ Q_0 = I,\ [Q_1, R_1] = qr(A) \text{ that is, } A = Q_1R_1 \\ [Q_{n+1}, R_{n+1}] & = qr(R_nQ_n) \\ \mathbf{y}_{n+1} & = Q'_{n+1}AQ_n\mathbf{y}_n + Q'_{n+1}\mathbf{b} \\ \mathbf{x}_{n+1} & = Q_{n+1}\mathbf{y}_{n+1}. \end{array} \right.$$

In the case of a constant – throughout the iterations – matrix A, one can recognize Francis' and Kublanovskaya's QR-algorithm. Using the convergence of (R_n) towards the matrix of eigenvalues of A (or towards its Schur form), in [10], Nedialkov and Jackson established the following bounds:

$$w(\mathbf{x}_n) \leq \text{cond}(P)\rho(A)^n w(\mathbf{x}_0) + \frac{\text{cond}(P)\rho(A)^{n-1} - 1}{\text{cond}(P)\rho(A) - 1} w(\mathbf{b}) + \mathbf{b}$$

where we recall A diagonalizable: $A = P^{-1}\Lambda P$.

2.5 Comparison

Two aspects are compared: the complexity and the accuracy, that is, the bounds on the widths of the iterates, of each method.

Let us first examine the computational complexity. Let us recall that the QR-factorization, resp. SVD-factorization, of a $d \times d$ matrix has a computational complexity of $\mathcal{O}(d^3)$. In the algorithms of Sections 2.3.2 and 2.3.3, the factorization of a matrix is performed only once, and not at every iteration: for n iterations, these algorithms thus have complexity $\mathcal{O}(d^3 + nd^2)$. In comparison, Lohner's QR method has complexity $\mathcal{O}(nd^3)$, which is significantly larger when d is large. In comparison, the cost of the factorization is negligible when the number n of iterations is large.

Let us now compare the accuracy of these different methods, from a theoretical point of view. The bounds we get on the width of the iterate $\mathbf{x}_n$ are larger than the bounds obtained by Nedialkov and Jackson, as the condition number of the diagonalizing matrix P appears to the n-th power in the formula for the QR- and SVD-algorithms, whereas it appears without this n-th power in the bound for Lohner's QR-algorithm. As a condition number is always larger or equal to 1, this means that the bound for Lohner's QR-algorithm is tighter than the bounds for the QR- and SVD-algorithms.

3 Experiments

3.1 Experimental Setup

After the results on the widths of the iterates of the toy example given in Section 1.1, Section 3.2 presents the computation of each corner of the initial box, to illustrate that it is possible to get tight enclosures, on such a small example.

All algorithms presented in this paper, namely the naive (or brute-force) application of the iteration, the QR-algorithm of Section 2.3.2, the two versions of the SVD-algorithm of Section 2.3.3, and Lohner's QR algorithm given in Section 2.4 have been implemented in Octave using Heimlich' interval package [4], then in Matlab using Rump's Intlab package [13]. Two other methods have been implemented and compared. The first technique [11] consists in the determination of k such that $\rho(|A^k|) < 1$, then it computes only one iterate every k step, in other words it computes

$$x_{(k+1)n} = A^k x_{kn} + \sum_{i=0}^{k-1} A^i b :$$

this iteration converges even when interval arithmetic is employed. The other technique is the use of affine arithmetic [3], as advocated by Rump in a private communication. The affine arithmetic employed here is the one available in Intlab [12]. In the experimental results presented below, each technique is associated to a color, as shown in Table 2.

Table 2: Color code for each method

algorithm	**color**
brute force	black
QR	cyan
SVD U	red
SVD V	magenta
Lohner's QR	dark blue
every k-th iterate	green
affine arithmetic	gray

The factorizations use only the basic QR and SVD factorizations available in Matlab, but neither the pivoted QR recommended by Lohner in [8] nor more elaborate versions presented by Higham in [5]. Very preliminary experiments with Lohner's pivoted QR factorization (not presented here) do not exhibit any difference in the width of divergence.

Sections 3.3 and 3.4 contain the evolution of the radii of the iterates computed by these different techniques, for two matrix dimensions: 10×10 and 100×100. The y-axis for the radii uses a logarithmic scale. For both dimensions, four kinds of matrices A have been used for the experiments. On the one hand, matrices which are well-conditioned (with a condition number of order 10^2) and ill-conditioned (with a condition number of order 10^{10}) have been generated. On the other hand, the scaling of the matrices varies: matrices which are well-scaled and matrices which are ill-scaled, with the order of magnitude of their coefficients varying between 1 and 10^{10}. More specifically, well-scaled matrices A are generated by a call to `randsvd` with the prescribed condition number. It is known that these matrices are well-scaled: all coefficients are of the same order of magnitude, they are said to be scaled, or equilibrated [6, Section 7.2]. Ill-scaled matrices are obtained as DAD^{-1} where the matrices A are generated by a call to `randsvd` as before and D is a diagonal matrix with diagonal elements varying in magnitude between 1 and 10^{10}: the coefficients of the resulting matrix also vary in magnitude, they are unscaled.

These denominations of "well-scaled" and "ill-scaled", "well-conditioned" and "ill-conditioned" are qualitative and not totally quantitative, as the matrices, originally generated by a call to Matlab's `randsvd` were then added to a multiple of the identity matrix and multiplied by a constant, in order to satisfy both $\rho(A) < 1$ and $\rho(|A|) > 1$. It can also be noted that degrading the scaling of the matrix also degrades its condition number; in other words, a "well-conditioned ill-scaled" matrix has a much worse condition number than a "well-conditioned well-scaled" matrix, even if the required condition numbers, in the call to `randsvd`, are initially the same.

Preliminary experiments with interval matrices, that is, matrices with interval coefficients, put into evidence two difficulties. First, the divergence is much faster:

with an interval matrix whose width is a thousandth of the center matrix, it takes less than 5 iterations to get a width that is several millions larger than the width of the iterates obtained using the center matrix, or even more. Second, accounting for interval matrices in the algorithms must be done with care: indeed, QR and SVD factorizations apply only to point matrices, thus they are used for the center matrices, but then the rest of the algorithm must correctly incorporate the interval parts of the matrices.

All experiments have been performed on a 2.7 GHz Quad-Core Intel Core i7 with 16GB RAM. Timings are averaged over 100 executions, except for affine arithmetic where at most 10 executions were performed.

3.2 Toy Example

First, the toy example presented in Section 1.1 is considered. As the iteration is affine, one can compute separately the images of the endpoints of the initial vector, to get the endpoints of the successive iterates. That is, we compute separately

$$x_n = 1.8 * x_{n-1} - 0.9 * x_{n-2} + 4.7.10^{-2} * 3 * \mathbf{u}$$

for $x_0 = 0$ and $x_1 = 1$ and for $x_0 = 0$ and $x_1 = 1.1$. However, we use the interval vector $\mathbf{u} = [9.95, 10.05]$ in the iteration. The convex hull of the 10 first iterates are represented on the left part of Figure 1. It is obvious that the width of the successive iterates grows rapidly.

Then we compute separately

$$x_n = 1.8 * x_{n-1} - 0.9 * x_{n-2} + 4.7.10^{-2} * 3 * u$$

for $x_0 = 0$ and $x_1 = 1$ and for $x_0 = 0$ and $x_1 = 1.1$, and for $u = 9.95$ and $u = 10.05$. The convex hull of the 10 first iterates are represented on the right part of Figure 1. In this case, the width of the successive iterates remains small, of the order of magnitude of 1% of the midpoint of the interval.

In this toy example, the iterations had to be performed 4 times, that is, once for each corner of the initial values, in order to get a tight enclosure. This can clearly not be generalized to high dimensions, as the number of corners grows as 4^d with the dimension d of the problem.

3.3 Example of dimension 10

Figure 2 gives the radii (in logarithmic scale) for the successive iterates computed by the methods detailed above. When the number of iterations is large (visually, above 30 or 40 iterations), the iterates computed by all methods presented in Section 2 diverge rapidly, as can be seen on the plots on the left. When one concentrates on the first iterations, the behaviours compare differently. One can also note that unscaling the matrix A speeds the divergence, for all methods. On the contrary, the k-step method and the use of affine arithmetic preserve the convergence of the iterates.

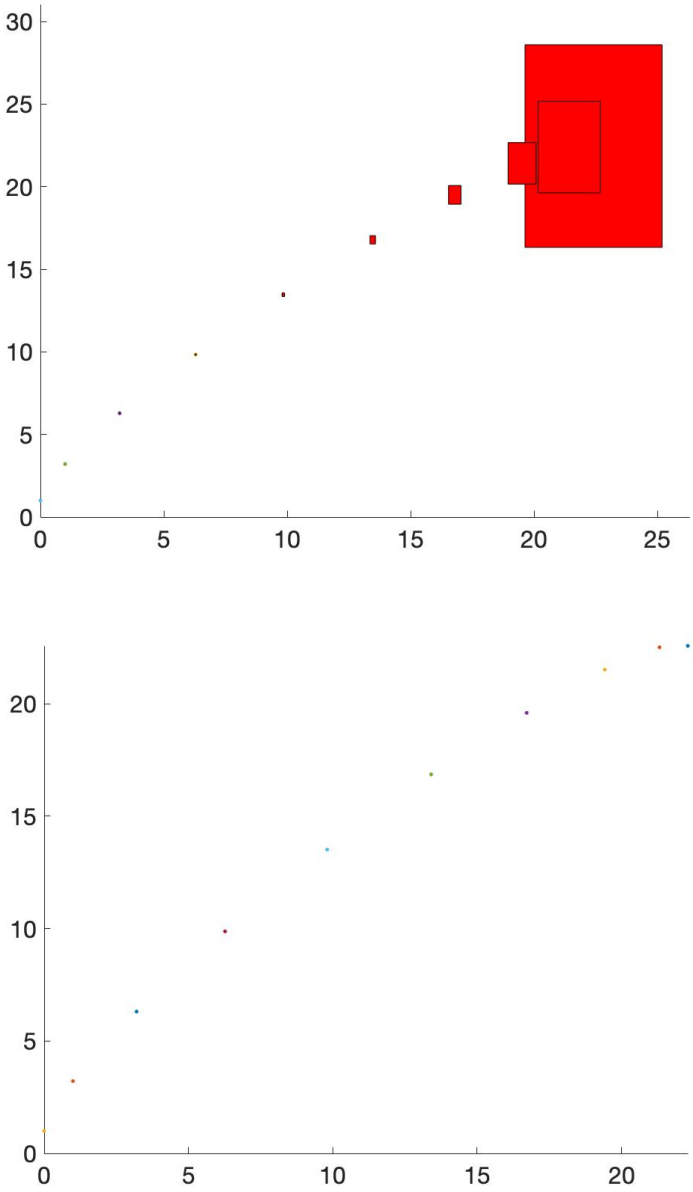
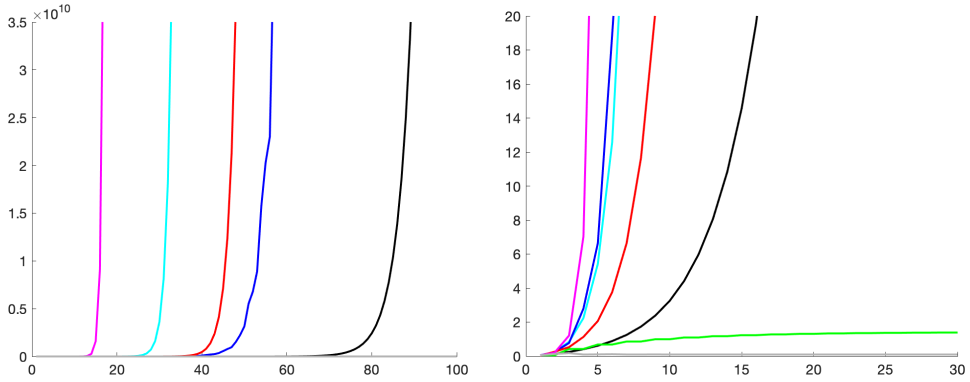
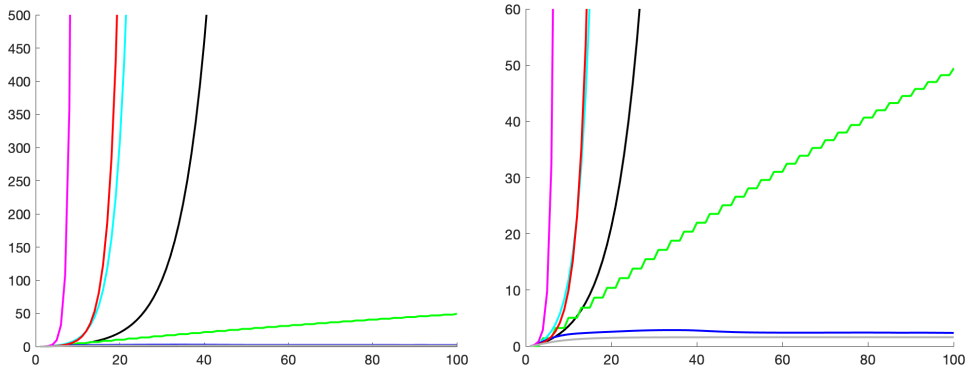


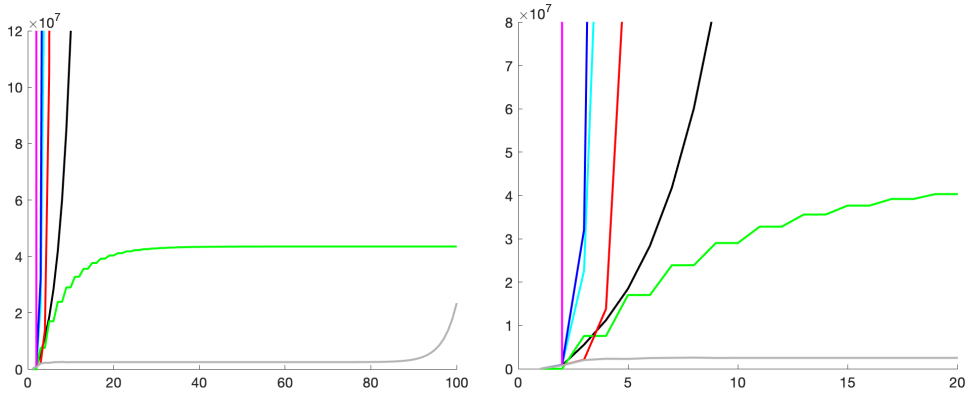
Figure 1: Top: the 10 first iterates of the toy example, where the endpoints of $\mathbf{x}_0$ are considered separately. Bottom: the 10 first iterates of the toy example, computed corner by corner.



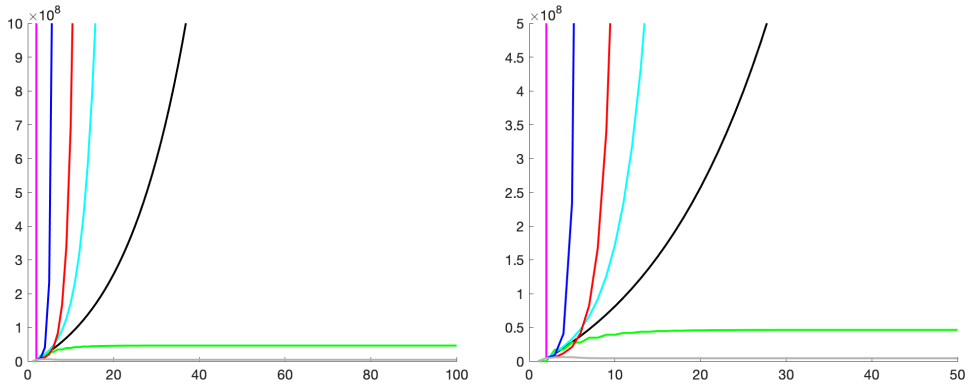
(a) Well-conditioned and well-scaled



(b) Ill-conditioned well-scaled



(c) Well-conditioned ill-scaled



(d) Ill-conditioned ill-scaled

Figure 2: The case of a 10×10 matrix (right part: zoom of the left part)

Table 3: Timings for 10×10 examples

method	**well-cond. well-scaled**	**ill-cond. well-scaled**	**well-cond. ill-scaled**	**ill-cond. ill-scaled**
naive	0.0088	0.0084	0.0093	0.0086
k-th step	0.0044	0.0015	0.0043	0.0045
QR	0.0161	0.0155	0.0160	0.0157
SVD U	0.0162	0.0156	0.0161	0.0166
SVD V	0.0147	0.0145	0.0324	0.0332
Lohner's QR	0.0170	0.0164	0.0165	0.0177
affine arith.	8.1859	8.7911	8.6595	8.2754

The timings in seconds are given in Table 3.

The methods presented in Sections 2.3.2, 2.3.3 and 2.4 all exhibit similar execution times. The naive method performs less operations and is thus faster. The k-th step method is fast as well, the variations in its execution time are due to the preprocessing, that is to the determination of the power k such that $\rho(|A^k|) < 1$: the execution time is larger when k is larger. With this method, the convergence is good. The use of affine arithmetic significantly slows down the computations, however the iterates converge.

3.4 Example of dimension 100

Figure 3 gives the radii (in logarithmic scale) for the successive iterates computed by the methods detailed in Section 3.1. When the number of iterations is large (visually, above 40 or 50 iterations), the iterates computed by all methods, except

the k-step method, diverge rapidly, as can be seen on the plots on the left. Again, when one concentrates on the first iterations, the behaviours compare differently.

The timings in seconds are given in Table 4.

Table 4: Timings for 100×100 examples

method	**well-cond. well-scaled**	**ill-cond. well-scaled**	**well-cond. ill-scaled**	**ill-cond. ill-scaled**
naive	0.0163	0.0145	0.0142	0.0142
k-th step	0.0046	0.0071	0.0048	0.0072
QR	0.1577	0.0363	0.0408	0.0409
SVD U	0.0427	0.0420	0.0437	0.0437
SVD V	0.0423	0.0390	0.0643	0.0638
Lohner's QR	0.0611	0.0758	0.0646	0.0804
affine arith.	70.8724	72.6441	76.6102	71.2518

The comments on the timings apply again, with the exception of the use of affine arithmetic, which is still much slower but does not manage any more to preserve the convergence very long. Indeed, the exponential growth applies when affine arithmetic is applied, as well as when other methods are employed; as the exponential growth occurs to the smaller terms (called the *noise*), affine arithmetic simply delays the divergence phenomenon.

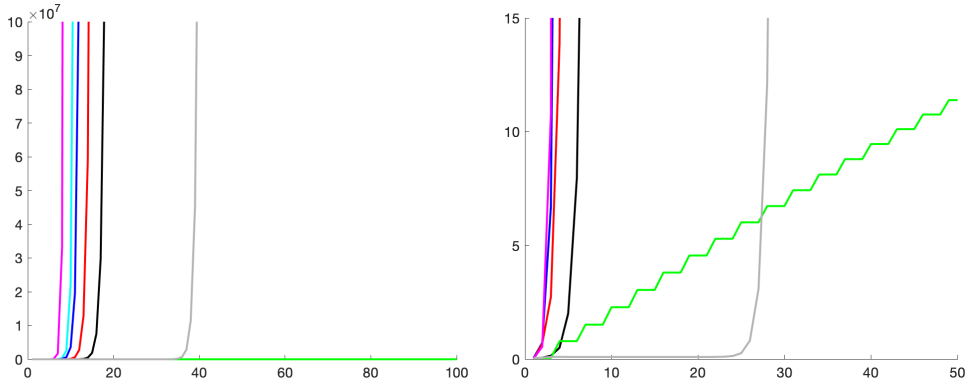
3.5 Comments

One can note that the k-step method, that is the method that resorts to a convergent interval iteration, performs very well at a moderate computation cost. Even the preprocessing time to determine the value of k has a negligible cost.

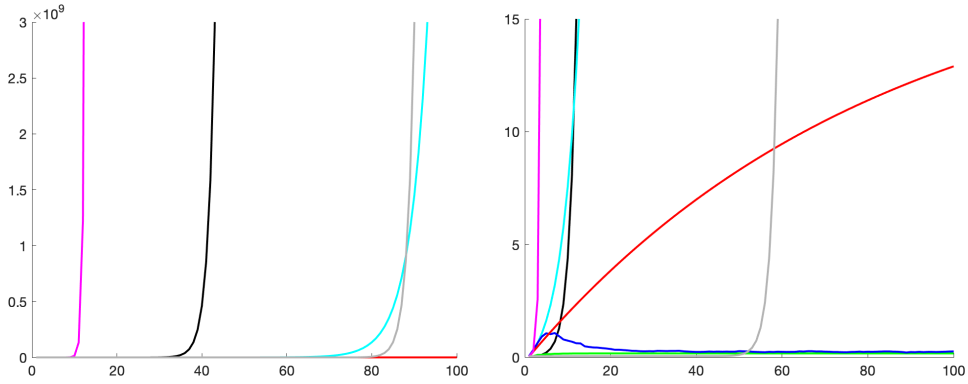
This method is a totally ad hoc approach for this problem and cannot be generalized. However, in the framework of filters and control theory, it has a physical meaning: the divergence of the iterations can be attributed to a sampling time which is too small to allow for significant variations to be observed. Multiplying the sampling time by k means sampling less frequently (by a factor k) and thus being able to measure the evolution of the observed quantities.

The use of affine arithmetic, on the contrary, is a very general method and it exhibits a very good accuracy, even if it eventually diverges (see the experiments with the 100×100 matrices in Section 3.4). The counterpart is the execution time, which is at least a thousand times larger than for the other methods. This is not an issue for the experiments presented here, as the time is of order of magnitude of a minute.

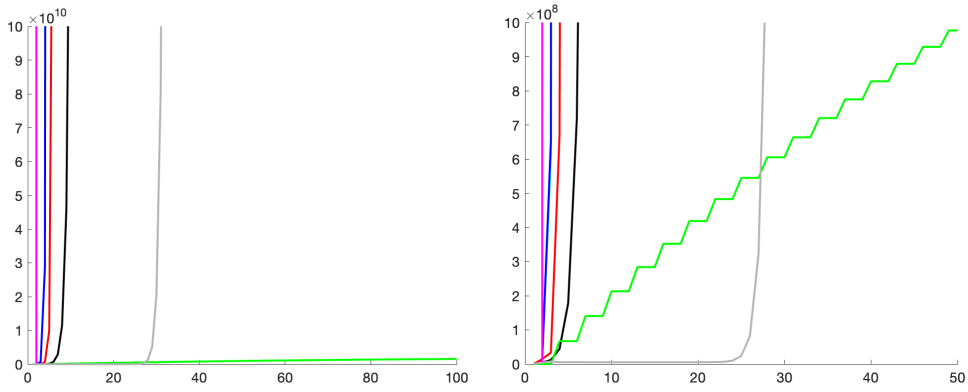
The methods based on the QR or SVD factorizations of the matrix A were developed with geometric principles in mind. For the QR-algorithm, the idea was to align the current box with the directions that are preserved by the product by A,



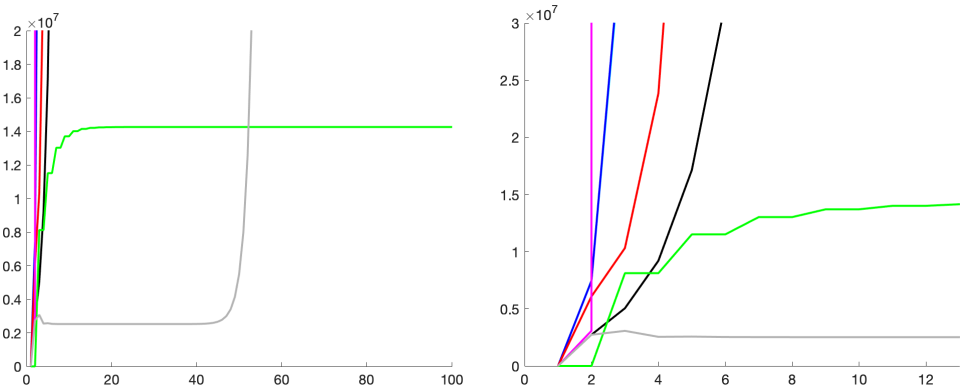
(a) Well-conditioned and well-scaled



(b) Ill-conditioned well-scaled



(c) Well-conditioned ill-scaled



(d) Ill-conditioned ill-scaled

Figure 3: The case of a 100×100 matrix (right part: zoom of the left part)

with a tradeoff between aligning the box along the eigenvectors and preserving an orthonormal system of coordinates, hence the choice of Q. For the SVD-algorithm, the idea was to align the box along the direction which gets the maximal elongation, that is along the singular vector corresponding to the largest singular value.

In both cases, the benefit of these geometric transformations is mitigated with the overestimation implied by extra computations, and there is either no clear benefit for the QR-based approach, or a delicate balance for the SVD-based approach. The SVD-algorithm is interesting when the matrix is ill-scaled, and particularly for the first iterations.

The methods of choice remain either the naive approach, when the matrix A is well-conditioned and well-scaled, or Lohner's QR method when the matrix is ill-conditioned. Surprisingly, the overhead of Lohner's QR method, in terms of computational time, is not as large as the formula for its complexity implies.

Our general recommendation is thus:

- to preprocess the matrix A in order to scale it (see [6, Section 7.3] about diagonal scaling);

- then to execute in parallel the naive approach and Lohner's QR approach, in order to converge reasonably well for any condition number of A.

Affine arithmetic is a solution of choice when other solutions fail and when the analysis and developing time is a scarce resource.

4 Conclusion and Future Work

This study, both theoretical and experimental, has compared several approaches to counteract the wrapping effect for the computation of affine iterations. Geometric considerations have led to the proposed algorithms. The benefit of these approaches is not always clear, as a better configuration is obtained through extra-computations and thus extra-overestimation. To deepen this geometric approach, we will aim at simplifying the resulting formulas, at getting formulas that are closer to the mathematically equivalent, but simpler, ones that are given after each proposed transformation. The main difficulty is to perform products such as $Q.Q'$ or $U'.U$, without replacing them by the identity, but in a certified and tight way. As the SVD-based approach seems more promising, our future work will concentrate on the use of a certified SVD factorization, as proposed by van der Hoeven and Yakoubsohn in [14]. We also plan to consider an interval version of the matrix, using the results in [7] to keep guarantees on the singular quantities involved in the computations.

References

[1] Arndt, H-R. and Mayer, G. On the semi-convergence of interval matrices. *Linear Algebra and its Applications*, 393:15–35, 2004. DOI: 10.1016/j.laa.2003.10.015.

[2] Beaumont, O. Solving interval linear systems with oblique boxes. Technical Report 1315, Irisa, 2000. ftp://ftp.irisa.fr/techreports/2000/PI-1313.ps.gz.

[3] De Figueiredo, L.H. and Stolfi, J. Self-validated numerical methods and applications. In *Monograph for 21st Brazilian Mathematics Colloquium, IMPA, Rio de Janeiro*, Volume 5, 1997.

[4] Heimlich, O. Interval arithmetic in GNU Octave. In *SWIM 2016: Summer Workshop on Interval Methods, France*, 2016. https://swim2016.sciencesconf.org/data/SWIM2016_book_of_abstracts.pdf#page=27.

[5] Higham, N.J. QR factorization with complete pivoting and accurate computation of the SVD. *Linear Algebra and its Applications*, 309:153–174, 2000. DOI: 10.1016/S0024-3795(99)00230-X.

[6] Higham, N.J. *Accuracy and Stability of Numerical Algorithms (2nd edition)*. SIAM, 2002. DOI: 10.1137/1.9780898718027.

[7] Hladik, M., Daney, D., and Tsigaridas, E. Bounds on real eigenvalues and singular values of interval matrices. *SIAM J. Matrix Analysis and Applications*, 31(4):2116–2129, 2010. DOI: 10.1137/090753991.

[8] Lohner, R. On the ubiquity of the wrapping effect in the computation of error bounds. In Kulisch, Lohner, Facius, editor, *Perspectives on Enclosures*

Methods, pages 201–217. Springer, 2001. DOI: 10.1007/978-3-7091-6282-8_12.

[9] Mayer, G. and Warnke, I. On the fixed points of the interval function $[f]([x]) = [a][x] + [b]$. *Linear Algebra and its Applications*, 363:202–216, 2003. DOI: 10.1016/S0024-3795(02)00254-9.

[10] Nedialkov, N. and Jackson, K. A new perspective on the wrapping effect in interval methods for initial value problems for ordinary differential equations. In Kulisch, Lohner, Facius, editor, *Perspectives on Enclosures Methods*, pages 219–264. Springer, 2001. DOI: 10.1007/978-3-7091-6282-8_13.

[11] Revol, N. Convergent linear recurrences (with scalar coefficients) with divergent interval simulations. In *SCAN: 11th GAMM – IMACS Int. Symp. on Scientific Computing, Computer Arithmetic, and Validated Numerics (Japan)*, 2004.

[12] Rump, Siegfried M. and Kashiwagi, Masahide. Implementation and improvements of affine arithmetic. *Nonlinear Theory and Its Applications, IEICE*, 6(3):341–359, 2015. DOI: 10.1587/nolta.6.341.

[13] Rump, S.M. INTLAB – INTerval LABoratory. In Csendes, Tibor, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. DOI: 10.1007/978-94-017-1247-7_7.

[14] van der Hoeven, J. and Yakoubsohn, J.-C. Certified Singular Value Decomposition. Technical report, Laboratoire d'informatique de l'École Polytechnique Palaiseau, 2018. https://hal.archives-ouvertes.fr/hal-01941987.

Contents

ISSN 0324—721 X (Print)
ISSN 2676—993 X (Online)

Editor-in-Chief: Tibor Csendes