A New Strategy for Exact Determination of the Joint Spectral Radius

Eine neue Strategie zur exakten Bestimmung des gemeinsamen Spektralradius
Zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Dissertation von Dipl.-Math. Claudia Möller aus Aachen
Tag der Einreichung: 04.02.2015, Tag der Prüfung: 29.05.2015
Darmstadt — D 17/2015

1. Gutachten: Prof. Dr. Ulrich Reif
2. Gutachten: Prof. Dr. Tomas Sauer

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik
Arbeitsgruppe Geometrie
und Approximation

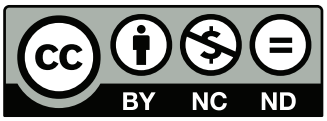
A New Strategy for Exact Determination of the Joint Spectral Radius
Eine neue Strategie zur exakten Bestimmung des gemeinsamen Spektralradius

Genehmigte Dissertation von Dipl.-Math. Claudia Möller aus Aachen

1. Gutachten: Prof. Dr. Ulrich Reif
2. Gutachten: Prof. Dr. Tomas Sauer

Tag der Einreichung: 04.02.2015
Tag der Prüfung: 29.05.2015

Darmstadt — D 17/2015



Abstract

Computing the joint spectral radius of a finite matrix family is, though interesting for many applications, a difficult problem. This work proposes a method for determining the exact value which is based on graph-theoretical ideas. In contrast to some other algorithms in the literature, the purpose of the approach is not to find an extremal norm for the matrix family. To validate that the finiteness property (FP) is satisfied for a certain matrix product, a tree is to be analyzed whose nodes code sets of matrix products. A sufficient, and in certain situations also necessary, criterion is given by existence of a finite tree with special properties, and an algorithm for searching such a tree is proposed. The suggested method applies in case of several FP-products as well and is not limited to asymptotically simple matrix families.
In the smoothness analysis of subdivision schemes, joint spectral radius determination is crucial to detect Hölder regularity. The palindromic symmetry of matrices, which results from symmetric binary subdivision, is considered in the context of set-valued trees.
Several illustrating examples explore the capabilities of the approach, consolidated by examples from subdivision.

Kurzzusammenfassung

Die Berechnung des gemeinsamen Spektralradius (*joint spectral radius*) einer endlichen Matrixfamilie ist, obgleich für viele Anwendungen interessant, ein schwieriges Problem. Diese Arbeit schlägt eine Methode zur Bestimmung des exakten Wertes vor, die auf graphentheoretischen Ideen basiert. Im Gegensatz zu einigen anderen Algorithmen aus der Literatur zielt dieser Ansatz nicht darauf ab, eine Extremalnorm zu finden. Um zu bestätigen, dass die Endlichkeitseigenschaft (*finiteness property*, kurz: FP) von einem gewissen Matrixprodukt erfüllt wird, wird ein Baum analysiert, dessen Knoten Mengen von Matrixprodukten kodieren. Eine hinreichende und in gewissen Situationen auch notwendige Bedingung ist durch Existenz eines endlichen Baumes mit speziellen Eigenschaften gegeben, und ein Algorithmus für die Suche nach einem solchen Baum wird präsentiert. Die dargestellte Methode gilt auch im Falle von mehreren FP-Produkten und ist nicht auf asymptotisch einfache Familien beschränkt.
Im Rahmen der Glattheitsanalyse von Subdivisionschemata ist die Bestimmung des gemeinsamen Spektralradius äußerst wichtig, um die Hölder-Regularität zu ermitteln. Die palindromische Symmetrie von Matrizen, die aus symmetrischer binärer Subdivision resultieren, wird im Kontext der mengenwertigen Bäume betrachtet.
Mit mehreren illustrierenden Beispielen werden die Möglichkeiten des Ansatzes erkundet und durch Beispiele der Subdivision ergänzt.

Acknowledgement

Foremost, I thank my supervisor Prof. Ulrich Reif for his inspiring ideas, for his open-mindedness to new thoughts, for many fruitful discussions and numerous helpful advices. It was a perfect balance of freedom and guidance. Moreover, I thank Prof. Tomas Sauer for acting as a referee and the interest he took in my work.

Furthermore, I thank the members of my working group for the relaxed and collegial atmosphere. They were always willing to discuss occuring problems, often asked the right questions, and shared their knowledge with me. Special thanks to Tobias Ewald, Nicole Lehmann and Nada Sissouno with whom I gladly shared the room.

Thanks to Jana Hintz and Ines Mantwill for proofreading my thesis, as well as to my husband Marco not only for the good tips concerning implementation and performance of the algorithm.

Contents

1 Introduction

The joint spectral radius determination has been a fascinating topic for this thesis. First, there is a broad range of applications, connecting various fields of mathematics to linear algebra. Second, although the concept of the joint spectral radius (JSR) generalizes the spectral radius of a single matrix in a natural way to a family of matrices, the theoretic results concerning computability are discouraging. Third, the notion of the JSR has an interesting story. Rota and Strang introduced the JSR in 1960 in a paper of only three pages [RS60]. "The notion seems to be useful enough in certain contexts to warrant the following elementary discussion," was one of the first sentences. These contexts were not specified, it says only that the notion developed in a course on matrix theory. There was no further research on the topic for almost 30 years until Daubechies and Lagarias rediscovered the concept 1992 for a paper on wavelets [DL92a]. Nowadays, the joint spectral radius is very popular and plays an important role not only in the context of refinable functions and subdivision, but also in problems of discrete mathematics, combinatorics, probability theory, ordinary differential equations and switched linear systems; see [Jun09, GZ09, PJB10] and references therein for an overview.

Consider a finite set $\mathscr{A} = \{A_1, ..., A_m\}$ of matrices in $\mathbb{C}^{d\times d}$. To deal with products of its elements, we introduce the sets

$$\mathscr{I}_0 := \{\emptyset\}, \quad \mathscr{I}_k := \{1, \dots, m\}^k, \quad \mathscr{I} := \bigcup_{k\in\mathbb{N}_0} \mathscr{I}_k,$$

of *completely positive index vectors* of length $k \in \mathbb{N}_0$ and arbitrary length, respectively. By contrast, an *index vector* may contain also negative entries, whose special meaning will be explained in Chapter 3.
For $k \in \mathbb{N}$, we define the matrix product

$$A_I := A_{i_k} \cdots A_{i_1}, \quad I = [i_1, \dots, i_k] \in \mathscr{I}_k.$$

Otherwise, if $k = 0$, let $A_\emptyset := \mathrm{Id}$ be the identity matrix. The length of the vector $I \in \mathscr{I}_k$ is denoted by $|I| := k$. That is, any index vector $I \in \mathscr{I}$ encodes a matrix product A_I with $|I|$ factors.
With $\|\cdot\|$ being a submultiplicative matrix norm, the joint spectral radius

$$\hat{\rho}(\mathscr{A}) := \limsup_{k\to\infty} \max_{I\in\mathscr{I}_k} \|A_I\|^{\frac{1}{k}}$$

characterizes the largest asymptotic growth of arbitrary products of $\mathscr{A}$ normalized by their length.

Denoting by ρ the standard spectral radius of a matrix, the so-called three-member-inequality, established in [BW92], states

$$\max_{I \in \mathscr{I}_k} \rho(A_I)^{\frac{1}{k}} \leq \hat{\rho}(\mathscr{A}) \leq \max_{I \in \mathscr{I}_k} \|A_I\|^{\frac{1}{k}} \qquad k \in \mathbb{N}. \tag{1.1}$$

It inspired many algorithms for approximating the JSR, as for example [Gri96, Mae96]. Since the lower and upper bounds converge, certain variants on breadth-first search up to some level k on the tree of all products of $\mathscr{A}$ lead to an approximation using any submultiplicative norm. However, these graph-theoretical approaches are computationally expensive when striving for high accuracy, since the number of products to consider grows exponentially.
Another line of research is the computation of a norm adapted to the matrix family to ensure fast convergence of the upper and lower bound of (1.1) in order to obtain approximation algorithms that are computationally less expensive. Some approaches in that spirit are presented in [BN05, Pro05, PJB10, AS98, BNT05, PJ08]. However, there are intrinsic limitations for the efficiency of approximation algorithms. It is shown in [BN05] that no algorithm for an approximation with relative error ϵ exists which is polynomial in both the dimension of the matrices d and $1/\epsilon$, unless P=NP. Approaches to an exact determination as developed in [GWZ05, GZ08, GZ09, GP13, JCG14] attempt to compute a norm such that the lower and upper bound coincide and therewith determine $\hat{\rho}(\mathscr{A})$. To be precise, these methods aim at finding a so-called extremal norm $\|\cdot\|_*$ satisfying

$$\max_{I \in \mathscr{I}_k} \rho(A_I)^{\frac{1}{k}} = \max_{I \in \mathscr{I}_1} \|A_I\|_* \tag{1.2}$$

for some finite k. The computational effort is shifted from the evaluation of upper and lower bounds for increasing k to the determination of the unit ball defining $\|\cdot\|_*$. In case of success, the exact value of the JSR is determined. Otherwise, some of these algorithms terminate with an approximation.
Achieving (1.2) requires the existence of a product $A_J, J \in \mathscr{I}_k$ with

$$\rho(A_J)^{\frac{1}{k}} = \hat{\rho}(\mathscr{A}) \tag{1.3}$$

for some $k \in \mathbb{N}$. We say that $\mathscr{A}$ has the finiteness property (FP) if such a J exists, and call J satisfying (1.3) an FP-product. In the literature, such a product is also called an optimal product or a spectrum maximizing product.
Any of the approaches to an exact determination mentioned above is based on the assumption that $\mathscr{A}$ has the finiteness property and aim to validate (1.3) for a presumed FP-product. [LW95] conjectured that any finite matrix family has the finiteness property, which was disproven non-constructively in [BM02, BTV03, Koz05], while an explicit counterexample is given in [HMST11]. Nevertheless, we do not know of an example resulting from applications that did not exhibit the FP.
This work presents a graph-theoretical approach instead of a norm computation in order to determinate the exact value of the JSR. Our method also attempts to establish the FP but is based on an arbitrary submultiplicative norm, which links this

strategy to graph-theoretical approximation. But the tree to perform the search on is different. The knots represent sets of matrices rather than single matrix products. This crucial idea potentially reduces the analysis of infinite sets of products to a study of finite subtrees. In particular, this aspect facilitates automated verification of the FP by computer programs. Furthermore, the number of products of a certain length that are to be considered is potentially reduced dramatically by proceeding from breath-first search to depth first search.
The set-valued tree approach accounts for finite complex matrix families and applies also in case of several FP-products. The theoretical foundation, being an essential part of this work, was published in [MR14]. We additionally present an algorithm for families of real matrices, which was implemented in MATLAB. To obtain a rigorous mathematical proof that a product satisfies (1.3), the method principally allows for performing the calculations analytically or by means of interval arithmetic.

This thesis is structured as follows: To introduce the topic, Chapter 2 clarifies notations and recalls some well-known facts. Chapter 3 develops the notion of set-valued trees and presents the theoretical results which connect these trees to JSR determination. A sufficient condition to characterize the situation $\hat{\rho}(\mathscr{A}) = 1$ is the existence of a so-called J-complete tree, which is a finite set-valued tree with certain properties. Furthermore, it is shown that the existence of such a tree with respect to an appropriate norm is also a necessary condition if the family is product bounded and has a spectral gap at 1.
The algorithmic search for a $\mathscr{J}$-complete tree and practical issues of implementation are discussed in Chapter 4, involving two different approaches to compute an upper bound for the infinite set of products coded by a node. The choice of norm is another issue to be considered. Although the strategy applies for any submultiplicative norm, the choice of norm has an impact on the shape of a $\mathscr{J}$-complete tree.
To put the set-valued tree method into context, Chapter 5 discusses the literature concerned with the problem of computability as well as other methods for approximation and exact determination.
Though the JSR plays an important role in various fields of mathematics, this work concentrates on its application in the context of smoothness analysis of subdivision schemes, being topic of Chapter 6. To be precise, our focus is put on the class of univariate, linear, stationary, uniform, compactly supported schemes, for which smoothness analysis is well understood. The corresponding results needed for our purposes are summarized. In particular, the subdivision matrices for schemes of arbitrary arity are derived and an explicit formula for the entries is provided. In case of symmetric binary subdivision, the subdivision matrices are palindromic, which leads to a symmetric situation in some sense. The consequences for the set-valued tree method are discussed: When choosing an adequate norm, the set-valued trees are symmetric, which reduces the computational effort. However, by symmetry of spectral radii, FP-products occur in pairs such that a spectral gap at 1 is in general not to expect. Therefore, the so-called palindromic transformation of the matrix family is developed, which, in certain situations, leaves the JSR un-

changed but induces a spectral gap at 1.
The capabilities and limitations of the method were investigated by applying the algorithm to illustrating examples, including subdivision matrices. The strategy was successful for matrix families with diverse properties. Chapter 7 presents the results. Conclusion and outlook are given in Chapter 8.

2 Basic facts

Several different notations are common in the literature to describe the setting of joint spectral radius determination. This chapter introduces the notation of this work and presents some basic facts.

2.1 Setting

We consider a finite set $\mathscr{A} = \{A_1, ..., A_m\}$ of matrices in $\mathbb{C}^{d\times d}$. To deal with products of its elements, we introduce the sets

$$\mathscr{I}_0 := \{\emptyset\}, \quad \mathscr{I}_k := \{1, \dots, m\}^k, \quad \mathscr{I} := \bigcup_{k\in\mathbb{N}_0} \mathscr{I}_k,$$

of *completely positive index vectors* of length $k \in \mathbb{N}_0$ and arbitrary length, respectively. By contrast, an *index vector* may contain also negative entries, whose special meaning will be explained in Chapter 3.
For $k \in \mathbb{N}$, we define the matrix product

$$A_I := A_{i_k} \cdots A_{i_1}, \quad I = [i_1, \dots, i_k] \in \mathscr{I}_k. \tag{2.1}$$

Otherwise, if $k = 0$, let $A_\emptyset := \mathrm{Id}$ be the identity matrix. The length of the vector $I \in \mathscr{I}_k$ is denoted by $|I| := k$. That is, any index vector $I \in \mathscr{I}$ encodes a matrix product A_I with $|I|$ factors.
Let $\|\cdot\|$ be a submultiplicative matrix norm. In this work, we assume throughout that the matrix family $\mathscr{A}$ is finite. So it is in particular bounded.

Definition 2.1 *The* joint spectral radius (JSR) *of $\mathscr{A}$ is defined as*

$$\hat{\rho}(\mathscr{A}) := \limsup_{k\to\infty} \max_{I\in\mathscr{I}_k} \|A_I\|^{\frac{1}{k}}.$$

As a consequence of Fekete's Lemma, the limit exists such that

$$\hat{\rho}(\mathscr{A}) = \lim_{k\to\infty} \max_{I\in\mathscr{I}_k} \|A_I\|^{\frac{1}{k}}, \tag{2.2}$$

see [Jun09] for a proof.
If $\mathscr{A}$ consists of a single matrix A, then

$$\hat{\rho}(\mathscr{A}) = \lim_{k\to\infty} \|A^k\|^{\frac{1}{k}}.$$

By Gelfand's formula, this equals the standard spectral radius ρ of A. Therewith, the JSR indeed is a generalization of the concept of the spectral radius.
The JSR is a property of a set of matrices, being independent of the norm due to norm equivalence: For two submultiplicative norms $\|\cdot\|_1, \|\cdot\|_2$ being related by

$$\alpha \cdot \|\cdot\|_1 \leq \|\cdot\|_2 \leq \beta \cdot \|\cdot\|_1,$$

for all $I \in \mathscr{I}_k$ holds that

$$\alpha^{\frac{1}{k}} \cdot \|A_I\|_1^{\frac{1}{k}} \leq \|A_I\|_2^{\frac{1}{k}} \leq \beta^{\frac{1}{k}} \cdot \|A_I\|_1^{\frac{1}{k}}.$$

With $\lim_{k\to\infty} \alpha^{\frac{1}{k}} = \lim_{k\to\infty} \beta^{\frac{1}{k}} = 1$ follows the claim.

2.2 The three-member-inequality

Daubechies and Lagarias showed in [DL92a] that upper and lower bounds on $\hat{\rho}(\mathscr{A})$ are given by the so-called three-member-inequality

$$\max_{I\in\mathscr{I}_k} \rho(A_I)^{\frac{1}{k}} \leq \hat{\rho}(\mathscr{A}) \leq \max_{I\in\mathscr{I}_k} \|A_I\|^{\frac{1}{k}}, \qquad k \in \mathbb{N}. \tag{2.3}$$

Further, they introduced the generalized spectral radius, which bases on the lower bound of (2.3),

$$\rho_*(\mathscr{A}) := \limsup_{k\to\infty} \max_{I\in\mathscr{I}_k} \rho(A_I)^{\frac{1}{k}}.$$

Berger and Wang showed in [BW92] that the equality

$$\rho_*(\mathscr{A}) = \hat{\rho}(\mathscr{A})$$

holds for bounded families $\mathscr{A}$.
As a consequence, evaluation of spectral radii and norms of matrix products allows in principle an arbitrarily good approximation of the JSR.

In some cases, the three-member-inequality leads to the exact value of the JSR, namely in cases when lower and upper bound coincide, as in the following example. Consider $\mathscr{A} = \{A_1, A_2, A_3\}$ with

$$A_1 = \begin{pmatrix} -1 & 0 \\ 3 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 3 & 0 \\ 0 & -1 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 0 & -1 \\ 0 & 3 \end{pmatrix}.$$

Due to $\rho(A_2) = 3$ and $\|A_1\|_\infty = \|A_2\|_\infty = \|A_3\|_\infty = 3$, it follows from the three-member-inequality that $\hat{\rho}(\mathscr{A}) = 3$.
This example results from smoothness analysis of subdivision schemes, see Chapter 6. $\mathscr{A}$ can be deduced from the linearization of the ternary median interpolating scheme presented in [XY05]. The result $\hat{\rho}(\mathscr{A}) = 3$ confirms that the maximal Hölder regularity of the linearized scheme is given by $\beta_* = 2 - \log_3(3) = 1$.

2.3 Irreducibility and product-boundedness

Definition 2.2 *The matrix family $\mathscr{A}$ is called* reducible *if the matrices $A_1, \ldots, A_m$ have a common invariant linear subspace, i.e., there exists an invertible Matrix T and an integer $d' < d$ such that, for $i = 1, \ldots, m$,*

$$TA_iT^{-1} = \begin{pmatrix} B_i & C_i \\ 0 & D_i \end{pmatrix} \tag{2.4}$$

with $D_i \in \mathbb{C}^{d' \times d'}$. Otherwise, $\mathscr{A}$ is said to be irreducible.

The problem of determining the JSR of a reducible family can always be transformed to a problem involving only irreducible families. A family with common invariant subspace can be split into lower-dimensional families, which can be analyzed separately. If (2.4) holds, then $\hat{\rho}(\mathscr{A}) = \max\{\hat{\rho}(\mathscr{B}), \hat{\rho}(\mathscr{D})\}$ with $\mathscr{B} := \{B_1, \ldots, B_m\}$ and $\mathscr{D} := \{D_1, \ldots, D_m\}$. See e.g. [Jun09] for a proof.

Nevertheless, we do not assume the matrix family to be irreducible unless indicated otherwise. Most of our results also hold for reducible families such that a check for irreducibility is not required. However, in practice, the splitting of a given problem might be very useful for its analysis.

Definition 2.3 *$\mathscr{A}$ is called* product bounded *if the set of all products $\{A_I : I \in \mathscr{I}\}$ is bounded.*

Product boundedness implies $\hat{\rho}(\mathscr{A}) \leq 1$. With $c_{\mathscr{A}}$ being the bounding constant,

$$\lim_{k\to\infty} \max_{I \in \mathscr{I}_k} \|A_I\|^{\frac{1}{k}} \leq \lim_{k\to\infty} c_{\mathscr{A}}^{\frac{1}{k}} = 1.$$

The following lemma ([Els95], Lemma 4) reveals the relation between the two properties.

Lemma 2.4 (Elsner) *If $\hat{\rho}(\mathscr{A}) = 1$ and $\mathscr{A}$ is not product bounded, then $\mathscr{A}$ is reducible.*

We will refer to this in Chapter 3.

2.4 Finiteness property

The set $\mathscr{A}$ is said to have the *finiteness property (FP)*, if there exists a completely positive index vector $J \in \mathscr{I}_k$ such that

$$\rho(A_J)^{\frac{1}{k}} = \hat{\rho}(\mathscr{A}). \tag{2.5}$$

A vector $J \in \mathscr{I}_k$ which satisfies (2.5) is called *FP-product*. The method suggested in this work as well as ideas developed in [GP13, CGSCZ10, GZ09, GZ08, Mae08, GWZ05, Mae00], are based on verifying (2.5) for some sophisticated guess J. A standard approach to find a candidate J is based on spotting repeating patterns in possibly long index vectors I which maximize either $\|A_I\|$ or $\rho(A_I)$.

The standard spectral radius as well as the JSR are homogeneous functions, i.e.,

$$\rho(\beta A_J) = |\beta|\,\rho(A_J), \quad \hat{\rho}(\beta\mathscr{A}) = |\beta|\,\hat{\rho}(\mathscr{A}), \quad \beta \in \mathbb{C}, \tag{2.6}$$

where $\beta\mathscr{A} := \{\beta A_1, \ldots, \beta A_m\}$. Hence, discarding the trivial case $\rho(A_J) = 0$, we can scale the family $\mathscr{A}$ such that at least one of the leading eigenvalues of A_J equals 1, and in particular $\rho(A_J) = 1$. Equation (2.5), which has to be demonstrated, then reads

$$\hat{\rho}(\mathscr{A}) = \rho(A_J) = 1. \tag{2.7}$$

By (2.3), this is possible only if

$$\rho(A_i) \le 1, \quad i = 1, \ldots, m, \tag{2.8}$$

so that we assume this property, throughout.

Given some index vector $J \in \mathscr{I}$, let $\pi(J)$ be the set of all cyclic permutations of J. Arbitrary repetitions of such vectors form the set

$$\Pi(J) := \{I^k : I \in \pi(J), k \in \mathbb{N}\}.$$

It is an elementary result of linear algebra that $\rho(A_{J'}) = \rho(A_J)$ for $J' \in \pi(J)$, see [Mae96] for a proof. In particular, if $\rho(A_J) = 1$, then also $\rho(A_{J'}) = 1$ for all $J' \in \Pi(J)$.

3 Set-valued tree method

This section presents the theoretical foundation for our approach to exact determination of the joint spectral radius of a finite family $\mathscr{A}$ of matrices from $\mathbb{C}^{d\times d}$. We already published most of these results in [MR14]. Nevertheless, they are an essential part of this thesis and form the basis for the results presented in the other chapters. Certainly, the exposition of facts as well as notations closely resemble those in [MR14]. Section 3.1 introduces the mathematical structures and notations that are convenient to formulate and prove our findings. The main result in Section 3.2 gives a sufficient condition to establish $\hat{\rho}(\mathscr{A}) = 1$ in terms of set-valued trees. This is also a necessary condition in certain situations as discussed in Section 3.3. In particular, irreducible families with a spectral gap at 1 as defined below can always be handled by the set-valued tree approach.

3.1 The concept of set-valued trees

As explained in Chapter 2, we aim at verifying some guess for the FP-product. Hence, we consider a matrix family $\mathscr{A} = \{A_1, \ldots, A_m\}$ satisfying (2.8) for which the normalized equation (2.7) shall be proven.

A matrix family having the FP possesses in general more than one FP-product. Trivial additional FP-products are given by cyclic permutations: If A_J is FP-product, then $A_{J'}$ is the same for any $J' \in \Pi(J)$. But there might exist more FP-products. We consider a family $\mathscr{J} \subset \mathscr{I}$ coding our candidates for FP-products. In order to show (2.7), candidates for FP-products are given by index vectors J with $\rho(A_J) = 1$. The existence of such a J is guaranteed by the scaling procedure. Moreover, for good reasons, we allow also index vectors with $\rho(A_J) < 1$. As will be demonstrated by an example in Section 7.1, such index vectors may reduce significantly the complexity of the trees to be constructed.

Definition 3.1 *Given matrices $\mathscr{A} = \{A_1, \ldots, A_m\}$, consider some non-empty set $\mathscr{J} = \{J_1, \ldots, J_n\}$ of completely positive index vectors $J_i \in \mathscr{I}$. If*

$$\max_{J\in\mathscr{J}} \rho(A_J) = 1,$$

then $\mathscr{J}$ is called a generator set *of $\mathscr{A}$, and each element $J \in \mathscr{J}$ is called a* generator *of $\mathscr{J}$. A generator J is called* strong *if $\rho(A_J) = 1$, and* weak *otherwise.*

That is, strong generators code candidates for FP-products while weak generators usually code matrix products with spectral radii close to 1. A generator set contains at least one strong generator.

It is our goal to relate properties of generator sets to the equality $\hat{\rho}(\mathscr{A}) = 1$. By (2.3), existence of a generator set implies $\hat{\rho}(\mathscr{A}) \geq 1$, so that (2.7) becomes equivalent to $\hat{\rho}(\mathscr{A}) \leq 1$. In the following, let the set $\mathscr{A} = \{A_1, \ldots, A_m\}$ of matrices and the generator set $\mathscr{J} = \{J_1, \ldots, J_n\}$ be fixed. Whenever a generator is mentioned, it is understood as a generator of $\mathscr{J}$. To address products of matrices in $\mathscr{A}$ conveniently, we introduce the sets

$$\mathscr{K}_0 := \{\emptyset\}, \quad \mathscr{K}_\ell := \{-n, \ldots, -1, 1, \ldots, m\}^\ell, \quad \mathscr{K} := \bigcup_{\ell \in \mathbb{N}_0} \mathscr{K}_\ell,$$

of *index vectors* of length $\ell \in \mathbb{N}_0$ and arbitrary length, respectively. As before, the length of $K \in \mathscr{K}_\ell$ is denoted by $|K| := \ell$.
Index vectors $K \in \mathscr{K}$ encode *sets* of matrix products in the following way: While single positive indices correspond to singletons, single negative indices correspond to infinite sets containing special matrix powers,

$$\mathscr{A}_i := \begin{cases} \{A_i\} & \text{if } i > 0, \\ \{A_{J_{-i}}^k : k \in \mathbb{N}_0\} & \text{if } i < 0. \end{cases}$$

Defining products of sets as sets of products, i.e., $\mathscr{P} \cdot \mathscr{Q} := \{PQ : P \in \mathscr{P}, Q \in \mathscr{Q}\}$, let

$$\mathscr{A}_K := \mathscr{A}_{k_\ell} \cdots \mathscr{A}_{k_1}, \quad K = [k_1, \ldots, k_\ell] \in \mathscr{K}_\ell,$$

for $\ell \in \mathbb{N}$, and $\mathscr{A}_\emptyset := \{\mathrm{Id}\}$. This definition is similar to (2.1), but A_I is a single matrix, while $\mathscr{A}_K$ is always a set, even if $K \in \mathscr{I}$ is completely positive. In this case, $\mathscr{A}_K = \{A_K\}$ is a singleton, while otherwise, it is typically[1] a denumerable set.
We need some more notations and definitions: *Concatenation* of vectors $P \in \mathscr{K}_h$ and $S \in \mathscr{K}_\ell$ is denoted by

$$[P, S] := [p_1, \ldots, p_h, s_1, \ldots, s_\ell] \in \mathscr{K}_{h+\ell}.$$

Powers indicate concatenation of an index vector with itself,

$$K^1 := K, \quad K^{\ell+1} := [K^\ell, K].$$

If $K = [P, S]$, then P is a *prefix* and S is a *suffix* of K. The sets of prefixes and suffixes of $K \in \mathscr{K}$ are denoted by

$$\mathscr{P}(K) := \{P : K = [P, S] \text{ for some } S\},$$
$$\mathscr{S}(K) := \{S : K = [P, S] \text{ for some } P\},$$

respectively.

[1] For instance, if all eigenvalues of the matrices A_J happen to be 0, then $\mathscr{A}_K$ is finite even if K is not completely positive.

Example 3.2 Let $\mathscr{J} = \{J_1, J_2\}$ with $J_1 = [1,2]$ and $J_2 = [2,2,1]$ be a generator set for a matrix set $\mathscr{A} = \{A_1, A_2\}$. We have $\mathscr{A}_{[-1]} = \{(A_2A_1)^k : k \in \mathbb{N}_0\}$ and $\mathscr{A}_{[-2]} = \{(A_1A_2^2)^k : k \in \mathbb{N}_0\}$. For $K = [1,1,-1,1,2]$, we obtain

$$\mathscr{A}_K = \{A_2A_1(A_2A_1)^kA_1^2 : k \in \mathbb{N}_0\}.$$

The first two entries of K generate the rightmost factor A_1^2, the negative entry -1 refers to the generator $J_1 = [1,2]$ and leads to the powers $(A_2A_1)^k$, and the last two entries generate the leftmost product A_2A_1. The index set $P = [1,1,-1] \in \mathscr{P}(K)$ is a prefix and $S = [1,2] \in \mathscr{S}(K)$ is the complementary suffix of K. In this special case, we observe that $\mathscr{A}_K \subset \mathscr{A}_P$, what might be surprising at first sight. This phenomenon, where a set of matrix products is completely covered by that of a prefix, will play a prominent role below.

In a natural way, the set $\mathscr{K}$ of index vectors can be given the structure of a directed *tree*, denoted by T: The elements of $\mathscr{K}$ are the nodes, the empty vector $\emptyset$ is the root, and an edge is connecting the *parent* node P with the *child* node C if and only if $C = [P,i]$ for some index $i \in \mathscr{K}_1$.

Definition 3.3 *A node $K \in \mathscr{K}$ is called*

- positive *or* negative *if so is the suffix $i \in \mathscr{K}_1$ when writing $K = [P,i]$.*
- 1-bounded *if* $\|\mathscr{A}_K\| := \sup\{\|A\| : A \in \mathscr{A}_K\} \le 1$.
- strictly 1-bounded *if* $\|\mathscr{A}_K\| := \sup\{\|A\| : A \in \mathscr{A}_K\} < 1$.
- covered *if there exists a prefix $P \in \mathscr{P}(K)$ such that $\mathscr{A}_K \subset \mathscr{A}_P$, and the complementary suffix S is completely positive and not empty.*

Typically, covered nodes appear in the following situation: Let $P = [P', \ell]$ be a negative node, i.e., $\ell < 0$. Then its descendant $K = [P, J_{-\ell}]$ is covered since

$$\mathscr{A}_{[\ell,J_{-\ell}]} = \mathscr{A}_{J_{-\ell}} \cdot \mathscr{A}_\ell = \{A_{J_{-\ell}}^{k+1} : k \in \mathbb{N}_0\} \subset \{A_{J_{-\ell}}^k : k \in \mathbb{N}_0\} = \mathscr{A}_\ell.$$

Example 3.2 is constructed in exactly this way. We call such a node $[P', \ell, J_{-\ell}]$ *combinatorially covered*.

The $(m+n)$-ary tree T depends only on the number m of matrices, and the number n of generators. The property of being positive or negative is independent of $\mathscr{A}$ and $\mathscr{J}$, too. To decide if a node is combinatorially covered, we need knowledge about $\mathscr{J}$ but not about $\mathscr{A}$. In contrast, the (strictly) 1-boundedness of a node depends on $\mathscr{A}$ and on the chosen norm.

Definition 3.4 *A finite subtree $T_* \subset T$ is called* $\mathcal{J}$-complete descending from K *if*

- *K is root of T_* and has m positive children*
- *every leaf of T_* is either* 1*-bounded or covered,*
- *every other node of T_* has either exactly m positive children or an arbitrary number of negative children.*

Such a tree is called minimal *if removing any of the nodes implies that at least one of the above conditions is not satisfied anymore.*

To shorten notation in the important particular case $K = \emptyset$, T_* is called $\mathcal{J}$-complete *if T_* is $\mathcal{J}$-complete descending from $\emptyset$.*

See Figure 3.1 for a visualization. Definition 3.4 can be extended without changes to the case $\mathcal{J} = \emptyset$, although, technically speaking, $\emptyset$ is not a generator set. Every node of a $\emptyset$-complete tree is completely positive and has either m positive children or is a 1-bounded leaf.

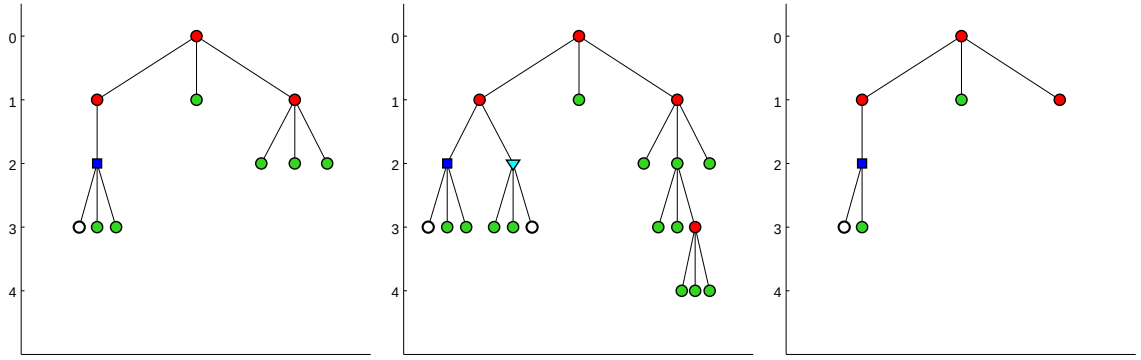


Figure 3.1: Visualization of $\mathcal{J}$-completeness for $m = 3$ and $\mathcal{J} = \{[1], [3]\}$. Colors and shapes of markers indicate properties of nodes: green $\triangleq$ 1-bounded, white $\triangleq$ covered, square $\triangleq$ negative child w.r.t. generator $J_1 = [1]$, triangle $\triangleq$ negative child w.r.t. generator $J_2 = [3]$, red $\triangleq$ other. The tree on the *left* is $\mathcal{J}$-complete and minimal. In contrast, the tree in the *middle* is $\mathcal{J}$-complete but not minimal since there is a node with more than one negative child and a 1-bounded node which is not a leaf. The tree on the *right* is not $\mathcal{J}$-complete because there is a leaf which is neither 1-bounded nor covered as well as a node with only 2 positive children.

A $\emptyset$-complete tree (descending from K) whose leaves are all *strictly* 1-bounded is called *contractive tree (descending from K)*. This type of tree characterizes the situation $\hat{\rho}(\mathcal{A}) < 1$ as shown in [HMR09]:

Lemma 3.5 *$\hat{\rho}(\mathcal{A}) < 1$ if and only if there exists a contractive subtree T_* of T.*

3.2 A sufficient condition for $\hat{\rho} = 1$

The following theorem provides a sufficient condition in terms of set-valued trees to establish (2.7) and therewith the JSR of the unscaled family. This condition is based on properties of a *finite* subtree of T and thus can be verified (though not falsified) numerically or analytically in finite time.

Theorem 3.6 *Let $\mathscr{A} = \{A_1, \dots, A_m\}$. If there exists a generator set $\mathscr{J}$ and a $\mathscr{J}$-complete tree $T_* \subset T$, then $\hat{\rho}(\mathscr{A}) = 1$.*

Algorithmic methods to validate this criterion for a given family are discussed in Chapter 4. To keep the computational effort as low as possible, such methods target a minimal $\mathscr{J}$-complete tree.

As explained in Section 2.3, we may assume that our matrix family is irreducible. But Theorem 3.6 is valid also in case of a reducible family. If T_* exists, then the products are either bounded or polynomially bounded as explained below.

The proof of Theorem 3.6 is based on the following ideas: By the existence of a generator set, it is $\hat{\rho}(\mathscr{A}) \geq 1$. To establish $\hat{\rho}(\mathscr{A}) \leq 1$, it suffices to show that there exists a monotone increasing polynomial $p : \mathbb{N}_0 \to \mathbb{R}$ such that all matrix products are bounded with respect to their length, $\|A_I\| \leq p(|I|)$. Convergence of $\sqrt[|I|]{p(|I|)}$ leads to the desired upper bound of $\hat{\rho}(\mathscr{A})$. To reach this goal, a partition for completely positive index vectors, depending on T_*, is defined. Lemmata 3.9 and 3.10 give upper bounds for the factors of this partition, together leading to the existence of p.

In the following, T_* is assumed to be a fixed subtree of T according to the conditions of the theorem. Dependencies of variables on T_* will not be declared explicitly. The finite set of nodes of T_* is denoted by $\mathscr{K}_*$, and the union of all contained matrix products by $\mathscr{A}_* := \bigcup_{K \in \mathscr{K}_*} \mathscr{A}_K$.
For any node $K \in \mathscr{K}_*$, there is $h \in \mathbb{N}_0$ such that

$$K = [I_1, -j_1, \dots, I_h, -j_h, I_{h+1}]$$

with $I_i \in \mathscr{I}$ for $1 \leq i \leq h+1$ and $j_{i_1} \in \{1, \dots, n\}$ for $1 \leq i \leq h+1$. We write

$$I \hookrightarrow K$$

if

$$I = [I_1, I_{J_{j_1}}^{k_1}, \dots, I_h, I_{J_{j_h}}^{k_h}, I_{h+1}]$$

for some $k_1, \dots, k_h \in \mathbb{N}_0$. Obviously, $A_I \in \mathscr{A}_K$ for $I \hookrightarrow K$.
Let

$$\mathscr{I}_* := \{I \in \mathscr{I} : \exists K \in \mathscr{K}_* \text{ with } I \hookrightarrow K\}.$$

The set $\mathscr{I}_*$ can be thought of as the set of completely positive index vectors corresponding to $\mathscr{A}_*$ since $\{A_I : I \in \mathscr{I}_*\} = \mathscr{A}_*$, and $\mathscr{K}_* \cap \mathscr{I} \subseteq \mathscr{I}_*$.

Example 3.7 Consider the tree with nodes $\mathscr{K}_* = \{\emptyset, [1], [2], [1,-1]\}$. With $J := J_1$, it is

$$\mathscr{A}_* = \{\mathrm{Id}, A_1, A_2\} \cup \{A_J^k A_1 : k \in \mathbb{N}_0\}$$

and

$$\mathscr{I}_* = \{\emptyset, [1], [2]\} \cup \{[1, J^k] : k \in \mathbb{N}_0\}.$$

Obviously, $\mathscr{I}_* \not\subset \mathscr{K}_*$ and $\mathscr{K}_* \cap \mathscr{I} = \{\emptyset, [1], [2]\} \subseteq \mathscr{I}_*$.

The index vectors in $\mathscr{K}$ are ordered completely by setting $K < K'$ for $|K| < |K'|$, and applying lexicographic order for index vectors of equal length. To any completely positive index vector $I \in \mathscr{I} \setminus \{\emptyset\}$, we assign the following two objects:

- The *$\mathscr{I}_*$-maximal prefix*

  $$M(I) := \max\{K \in \mathscr{P}(I) : K \in \mathscr{I}_*\}$$

  of I is defined as the longest prefix of I which is contained in $\mathscr{I}_*$.

- The *$\mathscr{K}_*$-maximal node*

  $$N(I) := \max\{K \in \mathscr{K}_* : M(I) \hookrightarrow K\}$$

  of I is defined as the maximal node in $\mathscr{K}_*$ with the property that $M(I) \hookrightarrow K$ for the $\mathscr{I}_*$-maximal prefix $M(I)$. The according set $\mathscr{A}_{N(I)}$ contains the product $A_{M(I)}$.

We note that, in general, $M(I)$ is not a node of T_* while $N(I)$ is by definition. On the other hand, $N(I)$ is generally not a prefix of I. Further, $M(I)$ is completely positive by definition, coding the singleton $\{A_{M(I)}\}$. In contrast, $N(I)$ may have negative entries, then coding a denumerable set of products which contains $A_{M(I)}$. If $I \in \mathscr{I}$ is a node of T_*, then $M(I) = I$.
Further, for any $I \in \mathscr{I} \setminus \{\emptyset\}$, the length of the $\mathscr{I}_*$-maximal prefix $M(I)$ is at least 1 because the first entry of I is necessarily a child of the root of T_*.
The following example illustrates these definitions.

Example 3.8 For a matrix family $\mathscr{A} = \{A_1, A_2\}$ with generator set $\mathscr{J} = \{[1,2]\}$, let T_* be the $\mathscr{J}$-complete tree visualized in Figure 3.2. We consider the completely positive index vectors $I_1 = [1,2]$, $I_2 = [1,2,1,2]$ and $I_3 = [2,1,1,1,2]$. I_1 and I_2 are elements of $\mathscr{I}_*$, I_1 is additionally a node of T_*. I_3 is neither a node of T_* nor an element of $\mathscr{I}_*$. Since $I_1 \in \mathscr{I}_*$, $M(I_1) = I_1$. In this case, $M(I_1)$ is a node of T_*. Nevertheless, $N(I_1) \neq M(I_1)$ since $M(I_1)$ is not maximal. Instead, $N(I_1) = [1,2,-1]$. In case of I_2, $M(I_2) = I_2$ such that $M(I_2)$ is not a node of T_*. But $[1,2,1,2] \hookrightarrow [1,2,-1]$ and $[1,2,1,2] \hookrightarrow [1,2,-1,1,2]$. Therefore, by maximality, $N(I_2) = [1,2,-1,1,2]$. The I_*-maximal prefix of I_3 is necessarily shorter than I_3 since $I_3 \notin \mathscr{I}$. Indeed, $M(I_3) = [2]$ and $N(I_3) = M(I_3)$. In Figure 3.2, the I_*-maximal nodes are labeled.

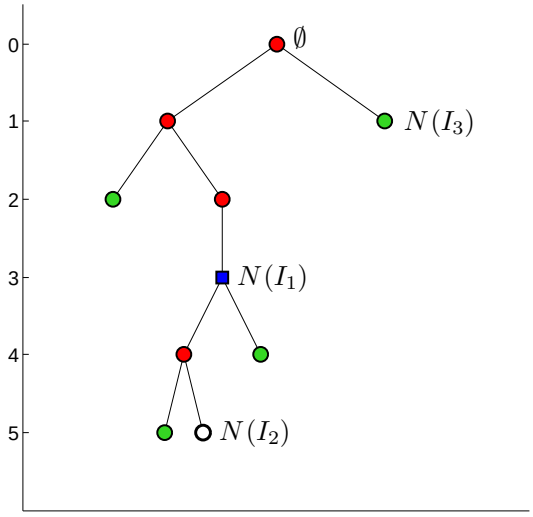


Figure 3.2: Exemplary tree to illustrate the definitions $M(I)$ and $N(I)$ for $I_1 = [1,2]$, $I_2 = [1,2,1,2]$ and $I_3 = [2,1,1,1,2]$. In this case, $\mathscr{A} = \{A_1, A_2\}$ and $\mathscr{J} = \{[1,2]\}$.

Lemma 3.9 *If $I \notin \mathscr{I}_*$, then its $\mathscr{K}_*$-maximal node $N(I)$ is a 1-bounded leaf of T_*.*

Proof. Let $P := M(I)$ and $K := N(I)$ denote the $\mathscr{I}_*$-maximal prefix and the $\mathscr{K}_*$-maximal node of I, respectively. First, assume that K has positive children. Writing $I = [i_1, \ldots, i_k]$ and $P = [i_1, \ldots, i_\ell]$, we know that $\ell + 1 \leq k$ because $I \notin \mathscr{I}_*$. Since, by assumption, positive children always come as a complete set of m siblings, $K' := [K, i_{\ell+1}] \in \mathscr{K}_*$ is a node of T_*. Consider the prefix $P' := [i_1, \ldots, i_{\ell+1}]$ of I. Then

$$A_{P'} = A_{i_{\ell+1}} \cdot A_P \in \mathscr{A}_{i_{\ell+1}} \cdot \mathscr{A}_K = \mathscr{A}_{K'} \subset \mathscr{A}_*.$$

This implies $P' \in \mathscr{I}_*$, contradicting maximality of the prefix P in $\mathscr{I}_*$.
Second, assume that K has a negative child $K' = [K, j]$ for some $j < 0$. Since $\mathscr{A}_j = \{A_{J_{-j}}^k : k \in \mathbb{N}_0\}$ contains the identity matrix, we have $A_P \in \mathscr{A}_K \subset \mathscr{A}_{K'}$, contradicting maximality of K.

Third, assume that K is a covered node. Thus, by definition, $K = [Q, S]$ for some $S \in \mathscr{I} \setminus \{\emptyset\}$ and $A_P \in \mathscr{A}_K \subseteq \mathscr{A}_Q$. By properties of S, Q has positive children. Again, the argument used in the first part of this proof yields a contradiction to maximality of the prefix P.

The first two cases imply that the node K is one of the leaves of T_*. These are either 1-bounded or covered, the latter being excluded by the third case. □

The *$\mathscr{I}_*$-maximal prefix partition* $P_1, \ldots, P_r$ of $I \in \mathscr{I} \setminus \{\emptyset\}$ is characterized by

$$\begin{aligned} I &= [P_1, \ldots, P_r] \\ P_\ell &= M([P_\ell, \ldots, P_r]), \quad \ell = 1, \ldots, r. \end{aligned}$$

Algorithmically, the vectors P_ℓ can be determined by a recursive process, starting from $P_1 = M(I)$: Regard the complementary suffix of I respective to prefix P_ℓ. Then $P_{\ell+1}$ is its $\mathscr{I}_*$-maximal prefix. The algorithm terminates finding a $\mathscr{I}_*$-maximal prefix $P_r = M(P_r)$, which implies $P_r \in \mathscr{I}_*$. The number r cannot exceed $|I|$ because $\mathscr{I}_*$-maximal prefixes of non-empty vectors have length ≥ 1. Lemma 3.9 provides information on the $\mathscr{I}_*$-maximal prefixes $P_1, \ldots, P_{r-1}$, while the suffix $S = P_r$ is covered by the following result:

Lemma 3.10 *There exists a monotone increasing polynomial $p : \mathbb{N}_0 \to \mathbb{R}$, depending only on $\mathscr{A}$ and T_*, such that*

$$\|A_S\| \leq p(|S|)$$

for any $S \in \mathscr{I}_*$.

Proof. Let $\mathscr{B} := \{A_1, \ldots, A_m, A_{J_1}, \ldots, A_{J_n}\}$. Since $\rho(B) \leq 1$ for all $B \in \mathscr{B}$, the entries of powers B^r grow at most in a polynomial way. That is, there exists a monotone increasing polynomial $q : \mathbb{N}_0 \to [1, \infty)$ with

$$\|B^r\| \leq q(r), \quad B \in \mathscr{B}, \quad r \in \mathbb{N}_0.$$

Let $h := \max\{|K| : K \in \mathscr{K}_*\}$ denote the depth of T_*. Then $p := q^h$ is also a monotone increasing polynomial on $\mathbb{N}_0$ and, due to the co-domain of q, $q^\ell \leq p$ for $\ell < h$. For $S \in \mathscr{I}_*$, the product A_S is a member of the set $\mathscr{A}_K$ for some $K = [k_1, \ldots, k_\ell] \in \mathscr{K}_*$, and thus can be written as

$$A_S = B_\ell^{r_\ell} \cdots B_1^{r_1}, \tag{3.1}$$

where $r_i \in \mathbb{N}_0, B_i \in \mathscr{B}$, and $B_i^{r_i} \in \mathscr{A}_{k_i}$.
The exponents are bounded by $r_i \leq |S|$, and the number of factors by $\ell \leq h$. Hence,

$$\|A_S\| \leq \prod_{i=1}^{\ell} \|B_i^{r_i}\| \leq \prod_{i=1}^{\ell} q(r_i) \leq q(|S|)^\ell \leq p(|S|),$$

as stated. □

The following example illustrates the notation in (3.1).

Example 3.11 Assume that $J_1 = [1, 2]$ and therewith $[1, 1, 2, 1, 2, 2] \hookrightarrow [1, -1, 2]$ such that $A_{[1,1,2,1,2,2]} \in \mathscr{A}_{[1,-1,2]}$. Then $B_1 = A_1 \in \mathscr{A}_1$, $r_1 = 1$, $B_2 = A_{J_1} = A_{[1,2]}$, $r_2 = 2$ and $B_3 = A_2 \in \mathscr{A}_2$, $r_3 = 1$. Also, $B_2^{r_2} = A_{J_1}^2 \in \mathscr{A}_{-1}$.

For a better understanding how the maximal prefix partition allows to apply lemmata 3.9 and 3.10, we provide another example.

Example 3.12 Consider T_* and $I_3 = [2,1,1,1,2]$ from Example 3.8. The $\mathscr{I}_*$-maximal prefix partition P_1, P_2, P_3 of I_3 is given by $P_1 = M([2,1,1,1,2]) = [2]$, $P_2 = M([1,1,1,2]) = [1,1]$ and $P_3 = M([1,2]) = [1,2]$. Since $[P_1,P_2,P_3] = [2,1,1,1,2] \notin \mathscr{I}_*$ and $[P_2,P_3] = [1,1,1,2] \notin \mathscr{I}_*$, the according $\mathscr{I}_*$-maximal nodes $N([2,1,1,1,2]) = [2]$ and $N([1,1,1,2]) = [1,1]$ are due to Lemma 3.9 1-bounded leaves of T_*, see also Figure 3.2. Since $P_3 = M(P_3) \in \mathscr{I}_*$, Lemma 3.10 states that $N(P_3) = [1,2,-1]$ is polynomially bounded.

Now, we are prepared to accomplish the proof of Theorem 3.6:
Proof. Let $I \in \mathscr{I}_k$ be any completely positive index vector, and $P_1, \dots, P_r$ its $\mathscr{I}_*$-maximal prefix partition. Then

$$A_I = A_{P_r} \cdot A_{P_{r-1}} \cdots A_{P_1}.$$

For $i = 1, \dots, r-1$, $[P_i, \dots, P_r] \notin \mathscr{I}_*$ and $A_{P_i} \in \mathscr{A}_{N([P_i,\dots,P_r])}$ by construction. So,

$$\|A_{P_i}\| \leq \|\mathscr{A}_{N([P_i,\dots,P_r])}\| \leq 1$$

due to Lemma 3.9.

A bound on the norm of P_r is given by Lemma 3.10. By monotonicity of the polynomial p and $|P_r| \leq |I| = k$, we obtain

$$\|A_I\| \leq \|A_{P_r}\| \leq p(|P_r|) \leq p(k).$$

Hence, by (2.3),

$$1 = \max\{\rho(A_J) : J \in \mathscr{J}\} \leq \hat{\rho}(\mathscr{A}) \leq \sqrt[k]{p(k)}.$$

As $k \to \infty$, the right hand side converges to 1, thus verifying the claim. □

We thus showed that the existence of a $\mathscr{J}$-complete tree validates $\hat{\rho} = 1$, a criterion which can be established algorithmically.

As mentioned before, there are two different types of covered nodes: There are nodes that are covered for any family $\mathscr{A}$ because of their combinatorial structure, being of the form $[P,-j,J_{-j}]$. Other covered nodes may occur by coincidence of products, which depends on the matrix family. Theorem 3.6 holds for arbitrary covered nodes. But the verification that a node is covered is difficult if it is not combinatorially covered. The implemented algorithm therefore identifies only combinatorially covered nodes. In the examples presented in Section 7, all covered nodes are of that type. We assume henceforth that leaves of a $\mathscr{J}$-bounded tree are 1-bounded or combinatorially covered.

In Definition 3.4, we request the root $\emptyset$ of T_* to have positive children for two reasons. First, $\emptyset$ is 1-bounded for any family $\mathscr{A}$ and we want to exclude the trivial tree that consists only of $\emptyset$. Second, positive children of the root simplified the proof of Theorem 3.6. We could equivalently demand that T_* has a positive node which is not the root:
Identifying a tree with its set of nodes, we consider

$$T_* = \{[-j, S] : S \in \mathscr{S}\}$$

for $j \in \{1, \dots, n\}$ and $\mathscr{S} \subset \mathscr{K}$ such that T_* is a minimal $\mathscr{J}$-complete tree descending from $[-j]$. We show that

$$\tilde{T}_* := \begin{cases} \mathscr{S} \cup \{[J_j, -j, S] : S \in \mathscr{S}\} & \text{if } J_j \in \mathscr{S}, \\ \mathscr{S} & \text{else} \end{cases}$$

is $\mathscr{J}$-complete.

If $J_j \notin \mathscr{S}$, all leaves $[-j, L]$ of T_* either 1-bounded, or covered such that $[-j, L] = [-j, P, -\tilde{j}, J_{\tilde{j}}]$ for some $J_{\tilde{j}} \in \mathscr{J}$,$P \in \mathscr{K}$. Since $L \hookrightarrow [-j, L]$, 1-bounded leaves of T_* induce 1-bounded leaves of $\mathscr{S}$. If $[-j, L]$ is covered, i.e., $L = [P, -\tilde{j}, J_{\tilde{j}}]$, then L is covered. Inheriting the combinatorial structure of T_*, the tree $\tilde{T}_* := \mathscr{S}$ is $\mathscr{J}$-complete.

If $J_j \in \mathscr{S}$, then T_* contains the covered node $[-j, J_j]$, but J_j is neither covered nor, in general, 1-bounded. Hence, $\mathscr{S}$ is not $\mathscr{J}$-complete in that case. By minimality of T_*, J_j is a leaf of $\mathscr{S}$. Since T_* is $\mathscr{J}$-complete descending from $[-j]$, no negative sibling but all m positive siblings of J_j are contained in $\mathscr{S}$. Therewith, any node of $\mathscr{S} \cup \{[J_j, -j, S] : S \in \mathscr{S}\}$ has either m positive children or an arbitrary number of negative children. The leaf $[J_j, -j, J_j]$ is covered. For all other leaves, the properties are inherited: Since $\mathscr{A}_{[J_j, L]} \in \mathscr{A}_{[J_j, -j, L]} \subset \mathscr{A}_{[-j, L]}$, the leaves $[J, L]$ and $[J, -j, L]$ are 1-bounded if the leaf $[-j, L]$ is 1-bounded. Considering a leaf $[-j, P, -\tilde{j}, J_{\tilde{j}}]$ of T_*, the according leaves $[P, -\tilde{j}, J_{\tilde{j}}]$ and $[J_j, -j, P, -\tilde{j}, J_{\tilde{j}}]$ are covered. This implies that $\tilde{T}_* = \mathscr{S} \cup \{[J_j, -j, S] : S \in \mathscr{S}\}$ is $\mathscr{J}$-complete, see Figure 3.3 for a visualization.

The described transformation from T_* to $\tilde{T}_*$ eliminates a negative node which has no positive prefix. It can be applied iteratively in order to achieve a tree whose root has positive children, and the number of steps is bounded due to finiteness of T_* . By requesting that T_* has at least one positive node which is not the root, we make sure that $\mathscr{S}$ is not empty, such that we terminate with a non-trivial tree $\tilde{T}_*$.

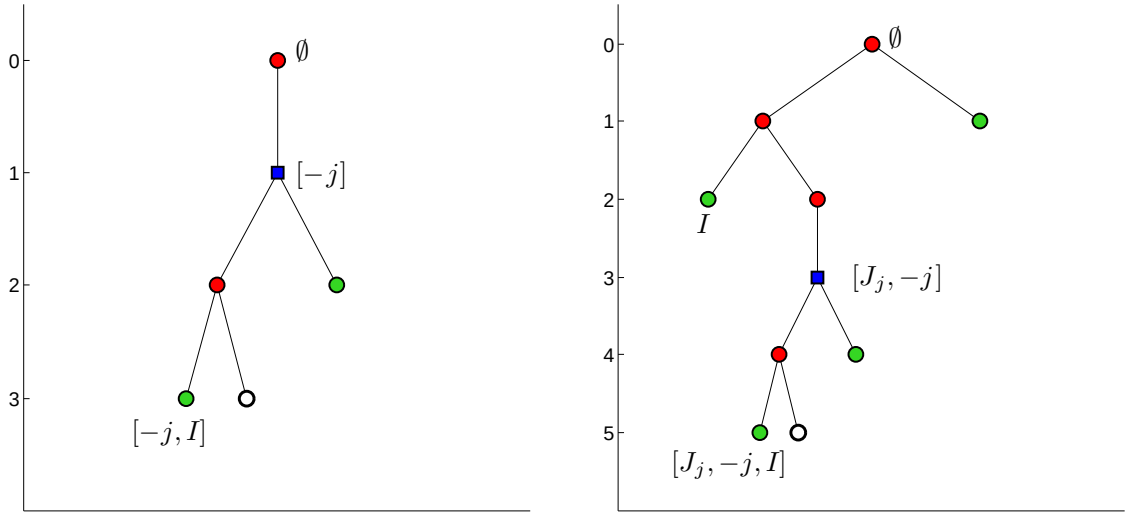


Figure 3.3: $\mathscr{J}$-complete tree descending from a negative child of the root *(left)* and a corresponding $\mathscr{J}$-complete tree *(right)* for $\mathscr{J} = \{[1,2]\}$. The tree on the left implies the existence of the one on the right.

Although including weak generators is often convenient, see Chapter 7, they are not essential for the existence of a $\mathscr{J}$-complete tree:

Proposition 3.13 *Let $\mathscr{J}$ be a generator set for a family $\mathscr{A}$ and*

$$\tilde{\mathscr{J}} := \{J \in \mathscr{J} : \rho(A_J) = 1\}$$

the subset of strong generators. If there exists a $\mathscr{J}$-complete tree T_*, then there exists a $\tilde{\mathscr{J}}$-complete tree $\tilde{T}_*$ as well.
Moreover, this still holds when requiring the 1*-bounded leaves of both trees to be strictly* 1*-bounded.*

Proof. Assume w.l.o.g. that T_* is minimal. By existence of $\mathscr{J}$, $\tilde{\mathscr{J}}$ is non-empty, and it is a generator set. Consider $J := J_j \in \mathscr{J} \setminus \tilde{\mathscr{J}}$. We show that any subtree of T_* with root $[P,-j]$ can be substituted such that the resulting tree is $\mathscr{J}$-complete. In the following, we identify a tree with its set of nodes. Let $\mathscr{S} \subset K$ such that the subtree of T_* with root $[P,-j]$ is given by

$$T_{\mathscr{S}} := \{[P,-j,S] : S \in \mathscr{S}\}.$$

Since T_* is $\mathscr{J}$-complete, $T_{\mathscr{S}}$ is a $\mathscr{J}$-complete tree descending from $[P,-j]$. We show that the tree

$$T_{\mathscr{S},\ell} := \bigcup_{0\le k\le \ell} \{[P,J^k,S] : S \in \mathscr{S}\}$$

is $\mathscr{J}$-complete descending from P for some $\ell \in \mathbb{N}_0$, based on the following preliminary considerations:
Since T_* is $\mathscr{J}$-complete, a node $S \in \mathscr{S}$ has either m positive children or 1 negative

child. If a leaf $[P,-j,S]$ of $T_{\mathscr{S}}$ is (strictly) 1-bounded, then $[P,J^k,S]$ is, for any $k \in \mathbb{N}_0$, a (strictly) 1-bounded leaf of $T_{\mathscr{S},\ell}$ since $[P,J^k,S] \hookrightarrow [P,-j,S]$. For a covered leaf $[P,-j,M,-\tilde{j},J_{\tilde{j}}]$ holds that $[P,J^k,M,-\tilde{j},J_{\tilde{j}}]$ is, for any $k \in \mathbb{N}_0$, a covered leaf of $T_{\mathscr{S},\ell}$.
We distinguish the following cases:
If $T_{\mathscr{S}}$ has no covered leaf $[P,-j,J_j]$ then $T_{\mathscr{S},0}$ is J-complete. Otherwise, $J_j \in \mathscr{S}$. This is a positive node, so all m positive siblings are contained in $\mathscr{S}$. Therewith, $T_{\mathscr{S},\ell}$, $\ell \in \mathbb{N}_0$ inherits of $\mathscr{S}$ that any node which is no leaf has either m positive children or 1 negative child. Since J is a weak generator, $\rho(A_J) < 1$. Hence, there exists $k_0 \in \mathbb{N}_0$ such that $\|A_J^{k_0}\| < 1$, i.e., $[P,J^{k_0}]$ is strictly 1-bounded. Together with the preliminary considerations, this implies that $T_{\mathscr{S},k_0-1}$ is $\mathscr{J}$-complete.
Since T_* is finite, only finitely many nodes $[P,-j]$, $P \in \mathscr{K}$, exist. Therewith, finitely many such transformations lead to a tree $\tilde{T}_*$ which contains no negative nodes corresponding to a weak generator. Then, $\tilde{T}_*$ is a $\tilde{\mathscr{J}}$-complete tree. □

3.3 Existence of a $\mathscr{J}$-complete tree in case of $\hat{\rho} = 1$

We will see that in some situations the existence of a $\mathscr{J}$-complete tree is not only a sufficient but also a necessary condition for $\hat{\rho} = 1$. This is the case, at least for a particular norm, if the spectral radii in the set of matrix products separate as described in the following.

Given a strong generator J, $\rho(A_J) = 1$ immediately implies $\rho(A_I) = 1$ for any index vector $I \in \Pi(J)$, see Section 2.3. A spectral gap at 1 means that the spectral radius of no other product matrix can come close to 1. More precisely, we define:

Definition 3.14 *The matrix family $\mathscr{A}$ has a* spectral gap at 1 *if there is exists a completely positive index vector J with $\rho(A_J) = 1$ such that*

- *there exists $q < 1$ such that*

$$\rho(A_I) \leq q \tag{3.2}$$

  for any product A_I, unless $I = \emptyset$ or $I = [S,J^r,P]$ for some $r \in \mathbb{N}_0$ and some partition $[P,S] = J$ of J,

- *the Jordan normal form Λ of A_J is*

$$\Lambda := V^{-1}A_JV = \begin{bmatrix} 1 & 0 \\ 0 & \Lambda_* \end{bmatrix}, \quad \rho(\Lambda_*) < 1. \tag{3.3}$$

In this case, J is called a dominant generator.

Recall that, since $\mathscr{A}$ is finite, the JSR equals the generalized spectral radius:

$$\hat{\rho}(\mathscr{A}) = \limsup_{k\to\infty} \max_{I\in\mathscr{I}_k} \rho(A_I)^{\frac{1}{k}}.$$

Therefore, a spectral gap at 1 implies $\hat{\rho}(\mathscr{A}) = 1$.

Assume that $\mathscr{A}$ is product bounded. That is, there exists a constant $c_{\mathscr{A}}$ such that $\|A_I\| < c_{\mathscr{A}}$ for all $I \in \mathscr{I}$. An algorithm for computing an admissible constant can be found in [Pro96]. If, in addition, $\mathscr{A}$ has a dominant generator J, we may scale the matrix V in (3.3) such that its columns v_j and the columns w_i of $W := V^{-\mathrm{t}}$ satisfy $\|w_1\|_2 = 1$ and $\|v_j\|_2 \le c_{\mathscr{A}}^{-1}, j \ge 2$, where the constant $c_{\mathscr{A}}$ is taken with respect to the Euclidean norm $\|\cdot\|_2$. Now, we define the matrix norm

$$\|A\|_V := \max_j \sum_i |w_i^{\mathrm{t}} A v_j|$$

as the standard 1-norm of the transformed matrix $W^{\mathrm{t}}AV$. At least with respect to this norm, the implication of Theorem 3.6 is in fact an equivalence:

Theorem 3.15 *Let $\mathscr{A}$ be product bounded with spectral gap at 1. Using the norm $\|\cdot\|_V$, there exists a tree $T_* \subset T$ according to the specifications of Theorem 3.6. Moreover, all nodes in this tree can be requested to be completely positive.*

Recall that due to Elsner's lemma (see Section 2.3), an irreducible family $\mathscr{A}$ with spectral gap at 1 always satisfies the condition of Theorem 3.15, and that we can always transform the problem such that irreducible matrices are to be analyzed. Therefore, demanding a spectral gap at 1 is the crucial condition.

To prove Theorem 3.15, we define the following: Let $I := (i_k)_{k\in\mathbb{N}}$ denote a sequence of positive indices $i_k \in \{1,\dots,m\}$. Adapting notation in the obvious way, we denote prefixes of I by $I_k := [i_1,\dots,i_k] \in \mathscr{P}(I)$. Further, $J^\infty := [J,J,\dots]$ is the sequence obtained by infinite repetition of J. If $\|A_{I_k}\| > 1$ for all $k \in \mathbb{N}_0$, then I is called an *infinite path*. The proof of Theorem 3.15 consists of three steps: First, in Lemma 3.16, we characterize the structure of such an infinite path. Second, in Lemma 3.17, we exclude the existence of an infinite path with respect to the V-norm $\|\cdot\|_V$. Third, we deduce the finiteness of a certain set-valued tree which satisfies the specifications of Theorem 3.6 and has only completely positive nodes.

Lemma 3.16 *Let J be a dominant generator of the product bounded family $\mathscr{A}$ with spectral gap at* 1. *Then any infinite path has the form $I = [P, J^\infty]$ for some prefix $P \in \mathscr{I}$.*

Proof. Let I be an infinite path. Since $\mathscr{A}$ is product bounded, the sequence $(A_{I_k})_k$ has a convergent subsequence $B_\ell := A_{I_{k(\ell)}}, \ell \in \mathbb{N}_0$, with limit $B^* := \lim_\ell B_\ell$. For any $\ell \in \mathbb{N}_0$ and $\lambda \in \mathbb{N}$ let $L_{\ell,\lambda} \in \mathscr{I}$ such that $\left[I_{k(\ell)}, L_{\ell,\lambda}\right] = I_{k(\ell+\lambda)}$. Then

$$B_{\ell+\lambda} = C_{\ell,\lambda} B_\ell, \quad C_{\ell,\lambda} := A_{L_{\ell,\lambda}}, \quad \ell \in \mathbb{N}_0.$$

The right hand side of

$$\|(C_{\ell,\lambda} - \mathrm{Id})B^*\| = \|C_{\ell,\lambda}(B^* - B_\ell) + B_{\ell+\lambda} - B^*\| \le c_{\mathscr{A}} \|B^* - B_\ell\| + \|B^* - B_{\ell+\lambda}\|$$

tends to 0 as $\ell \to \infty$. Thus,

$$\lim_{\ell\to\infty} C_{\ell,\lambda} B^* = B^*.$$

By assumption, $\|B_\ell\| > 1$ for all ℓ and hence $B^* \neq 0$. So, recalling (3.2), the displayed equation shows that there exists $\ell_{0,\lambda} \in \mathbb{N}$ such that $\rho(C_{\ell,\lambda}) > q$ for all $\ell \ge \ell_{0,\lambda}$. Since J is dominant, we obtain by definition

$$L_{\ell,\lambda} = S_{\ell,\lambda} J^{r_{\ell,\lambda}} P_{\ell,\lambda}, \quad \ell \ge \ell_{0,\lambda},$$

for partitions $[P_{\ell,\lambda}, S_{\ell,\lambda}] = J$. Substituting this representation into $[L_{\ell,1}, L_{\ell+1,1}] = L_{\ell,2}$ yields

$$S_{\ell,1} J^{r_{\ell,1}} P_{\ell,1} S_{\ell+1,1} J^{r_{\ell+1,1}} P_{\ell+1,1} = S_{\ell,2} J^{r_{\ell,2}} P_{\ell,2}$$

for $\ell \ge \ell_0 := \max\{\ell_{0,1}, \ell_{0,2}\}$. Since $|S_{\ell,2}| \le |J|$ and $|P_{\ell,2}| \le |J|$, this is possible only if $P_{\ell,1} S_{\ell+1,1} = J$, implying $P_{\ell,1} = P_{\ell+1,1}$ and $S_{\ell,1} = S_{\ell+1,1}$. That is, $P_{\ell,1} = P_{\ell_0,1}$ and $S_{\ell,1} = S_{\ell_0,1}$ for all $\ell \ge \ell_0$.

By definition of $L_{\ell,1}$, the infinite path is given for any $\ell \in \mathbb{N}_0$ by

$$I = [I_{k(\ell)}, L_{\ell,1}, L_{\ell+1,1}, L_{\ell+2,1} \dots]$$

We abbreviate $r_i := r_{\ell_0+i,1}$, $\tilde{P} := P_{\ell_0,1}$, $\tilde{S} := S_{\ell_0,1}$ to find

$$I = [I_{k(\ell_0)}, \tilde{S}, J^{r_0}, \tilde{P}, \tilde{S}, J^{r_1}, \tilde{P}, \dots] = [I_{k(\ell_0)}, \tilde{S}, J^\infty],$$

and the claim follows with $P := [I_{k(\ell_0)}, \tilde{S}]$. □

While the lemma above makes no assumptions concerning the underlying norm, the next one shows that the V-norm $\|\cdot\|_V$ is special.

Lemma 3.17 *If $\mathscr{A}$ is product bounded with spectral gap at 1, then there is no infinite path with respect to the V-norm.*

Proof. Assume that there exists an infinite path. According to the previous lemma, it is given by $I = [P, J^\infty]$ with J being a dominant generator. The corresponding limit of matrix products $S := \lim_k A_{[P,J^k]}$ exists because $\Lambda^\infty := \lim_k \Lambda^k = \operatorname{diag}([1, 0, \dots, 0])$ exists. Since $\|A_{J^k}\|_V = \|\Lambda^k\|_1 = 1$ for k sufficiently large, P cannot be empty. In this case, we know that $\|A_{[P,J^k]}\|_V > 1$ and $\rho(A_{[P,J^k]}) \leq q < 1$ for all $k \in \mathbb{N}$. Hence, $\|S\|_V \geq 1$ and $\rho(S) \leq q$. With $T := A_P$, the matrix $\tilde{S} := W^\mathrm{t} S V$ is given by

$$\tilde{S} = \Lambda^\infty W^\mathrm{t} T V = \begin{bmatrix} w_1^\mathrm{t} T v_1 & w_1^\mathrm{t} T v_2 & \cdots & w_1^\mathrm{t} T v_d \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Since S and $\tilde{S}$ are similar, we have $|w_1^\mathrm{t} T v_1| = \rho(\tilde{S}) = \rho(S) < 1$. Further, recalling $\|w_1\|_2 = 1$ and $\|v_j\|_2 \leq c_{\mathscr{A}}^{-1}$, we find $|w_1^\mathrm{t} T v_j| \leq \|w_1^\mathrm{t}\|_2 \|T\|_2 \|v_j\|_2 < 1$ for $j \geq 2$. Thus, we obtain the contradiction $1 \leq \|S\|_V = \|\tilde{S}\|_1 < 1$. □

The lemma enables us to prove Theorem 3.15:

Proof. Let T_* be the largest subtree of T with the following properties: $\emptyset$ is contained as a node, all nodes are completely positive and each 1-bounded node is a leaf. Assuming that the set $\mathscr{K}_*$ of nodes is infinite, we define $\mathscr{K}' \subset \mathscr{K}_*$ as the set of nodes which are prefix of infinitely many other nodes in $\mathscr{K}_*$. Then $\mathscr{K}'$ is not empty because the root $\emptyset$ belongs to it. Further, if $K \in \mathscr{K}'$, then there exists at least one child $[K, k] \in \mathscr{K}'$. That is, the recursion

$$i_k := \min\{i \in \mathscr{I}_1 : [i_1, \dots, i_{k-1}, i] \in \mathscr{K}'\}$$

defines an infinite path $I = (i_k)_{k \in \mathbb{N}}$, contradicting Lemma 3.17. □

Hence, an irreducible family with a spectral gap at 1 always possesses a J-complete tree with respect to $\|\cdot\|_V$, even when requesting all nodes to be positive. On the premise that the family has these properties, the search for a J-complete tree is very simple: A depth-first search on the tree with nodes $\mathscr{I}$ is guaranteed to terminate when backtracking in nodes which are 1-bounded w.r.t. $\|\cdot\|_V$. The class of irreducible families with a spectral gap at 1 is stressed also by other authors. Therewith, the result is important for comparison of the set-valued tree approach and other methods in the literature, see also Chapter 5. While termination is guaranteed for this class of families, it is not excluded for others. In fact, the set-valued tree strategy has proven to useful for families with most diverse properties, see Section 7 for examples.

Despite the theoretical importance, the practical value can be questioned. Theorem 3.15 states the existence of a $\mathscr{J}$-complete and therewith finite tree but gives

no bound on its depth. Furthermore, the $\mathscr{J}$-complete tree with respect to $\|\cdot\|_V$ is not necessarily trim. In that regard, other norms are potentially preferable.
Although requesting all nodes to be positive simplifies the search, the numerical realization is difficult since this involves checks on equality. This can be avoided when allowing a single negative node in the tree. Details on the algorithmic search are discussed in Section 4.4.

The following result concerns the stability of a $\mathscr{J}$-complete tree under small perturbations.

Proposition 3.18 *Consider a family $\mathscr{A}^\omega$ which continuously depends on $\omega \in \mathbb{R}$. Assume that $\mathscr{A}^0$ has a spectral gap at 1 with dominant generator J, and $\mathscr{J} = \{J\}$. Let T_* be a $\mathscr{J}$-complete tree for $\mathscr{A}^0$ whose leaves are either strictly 1-bounded or covered.*
Then there exists $\varepsilon > 0$ such that T_* is $\mathscr{J}$-complete for $\mathscr{B}^\omega := \frac{1}{\rho(A_J^\omega)^{|J|}}\mathscr{A}^\omega$ with $|\omega| \le \varepsilon$.

Proof. By assumption, A_J^0 is FP-product of $\mathscr{A}^0$. Since the spectral radius is a continuous function of the matrix and the family has a spectral gap at 1, A_J^ω is FP-product of $\mathscr{A}^\omega$ for $|\omega|$ sufficiently small. There exists $\omega_* > 0$ such that J is also dominant generator of $\mathscr{B}^\omega$ for $|\omega| \le \omega_*$. The property of a node to be covered[2] does not depend on the matrix family. To show that T_* is $\mathscr{J}$-complete for $\mathscr{B}^\omega$, it suffices to prove that leaves which are strictly 1-bounded for $\mathscr{A}^0 = \mathscr{B}^0$ are 1-bounded for $\mathscr{B}^\omega$.
Let K be a leaf of T_* which is not covered. If K is completely positive, then

$$\|\mathscr{B}_K^\omega\| = \|B_K^\omega\| = \frac{1}{\rho(A_J^\omega)^{|J|}}\|A_K^\omega\|.$$

Since $\frac{1}{\rho(A_J^\omega)^{|J|}}\|A_K^\omega\|$ is continuous in ω and $\|A_K^0\| = \frac{1}{\rho(A_J^0)^{|J|}}\|A_K^0\| < 1$, there exists ω_K such that K is 1-bounded w.r.t. $\mathscr{B}^\omega$ for $|\omega| \le \omega_K$.
Assume that K has one negative entry, that is, $K = [P, -1, S]$ for some $P, S \in \mathscr{I}$. Denote by λ_ω the subdominant eigenvalue of B_J^ω. Let $\mu := \frac{1}{2} + \frac{1}{2}\max_{\omega\in[-\omega_*,\omega_*]}|\lambda_\omega|$. By dominance of J on $[-\omega_*, \omega_*]$, $(B_J^\omega)^k = T_\omega + o(\mu^k)$, and T_ω depends continuously of ω. Then

$$\begin{aligned}\|B^\omega_{[P,J^k,S]}\| &= \|B_S^\omega \cdot (B_J^\omega)^k \cdot B_P^\omega\| \\ &= \|B_S^\omega \cdot T_\omega \cdot B_P^\omega + B_S^\omega \cdot o(\mu^k) \cdot B_P^\omega\| \\ &\le \|B_S^\omega \cdot T_\omega \cdot B_P^\omega\| + o(\mu^k) \qquad (3.4)\end{aligned}$$

Since $\|\mathscr{B}^0_{[P,-1,S]}\| = \sup_{k\in\mathbb{N}_0}\|B^0_{[P,J^k,S]}\| < 1$, and $\lim_{k\to\infty} B^0_{[P,J^k,S]} = B_S^0 \cdot T_0 \cdot B_P^0$, it holds that

$$\|B_S^\omega \cdot T_\omega B_P^\omega\| < 1$$

[2] We assumed that covered leaves are combinatorially covered, see Section 3.2.

for $|\omega|$ sufficiently small. Since $\mu < 1$, there exists $k_0 \in \mathbb{N}_0$ such that (3.4) is bounded by 1.
For the finite number of cases $k < k_0$, the arguments for completely positive nodes apply. Hence, there exists ω_K such that 1-boundedness is given for both $k < k_0$ and $k \geq k_0$ on $[-\omega_K, \omega_K]$, which implies $\|\mathscr{B}_K^\omega\| \leq 1$ for $|\omega| < \omega_K$.

Analogous argumentation applies for a node K with $\ell \leq |K|$ negative entries, and $|K|$ is bounded by the depth of T_*. Since the number of leaves is finite,

$$\omega_T := \min\{\omega_K : K \text{ is strictly 1-bounded leaf of } T_*\}$$

exists. Since we assumed all non-covered leaves to be strictly 1-bounded w.r.t. $\mathscr{A}^0$, the tree T_* is $\mathscr{J}$-complete w.r.t. $\mathscr{B}^\omega$ for $|\omega| \leq \min\{\omega_*, \omega_T\}$. □

We conjecture that a similar result holds for a family which has no spectral gap at 1. The scaled family $\mathscr{B}^\omega$ has to satisfy that $\mathscr{J}$ is a generator set for $\mathscr{B}^\omega$, that is,

$$\mathscr{B}^\omega := \frac{1}{\max_{J\in\mathscr{J}} \rho(A_J^\omega)^{|J|}} \mathscr{A}^\omega.$$

For I completely positive, B_I depends continuously of ω such that all arguments for completely positive nodes are similar. The proof that a leaf K with a negative entry, being strictly 1-bounded w.r.t. $\mathscr{A}^0$, is 1-bounded for $\mathscr{B}^\omega$ with $|\omega|$ sufficiently small, becomes more complex, though.

4 Algorithm

We established a theoretical connection between JSR and set-valued trees in Section 3.2, including a sufficient condition involving $\mathscr{J}$-complete trees. Nevertheless, it remains the question how to prove the existence of a $\mathscr{J}$-complete tree for fixed $\mathscr{A}$ and $\mathscr{J}$. This section presents an algorithm for finding such a tree based on depth first search. In Section 4.1, the theoretical framework without consideration of practical aspects is presented. These issues are discussed in Section 4.2, and in particular the question of how to check if a node is 1-bounded. Two approaches to this concern are developed in Section 4.3. The very important role of the underlying norm is topic of Section 4.4. A variant of the algorithm can be used to verify $\hat{\rho}(\mathscr{A}) < 1$ by searching for a contractive tree, see Section 4.5. Part of this work is an implementation of both algorithms in MATLAB. The intention was to give a proof-of-concept as well as to better understand the importance of the different parameters. Therefore, the implementation is not optimized in terms of efficiency or runtime. Nevertheless, a range of examples was completed successfully, most of them on a time-scale of seconds or minutes on a standard PC, see also Chapter 7.

4.1 The algorithm in theory

Let $\tilde{T}$ be an infinite subtree of T such that $\emptyset$ is root of $\tilde{T}$ having m positive children and that every other node of $\tilde{T}$ has either exactly m positive children or exactly one negative child.
Traverse $\tilde{T}$ by depth-first search and backtrack if the current node is 1-bounded or covered[1]. If the search terminates, the visited subtree of $\tilde{T}$ is obviously $\mathscr{J}$-complete. The shape of the tree depends on the chosen norm and there is no a priori bound for its depth or the number of nodes.
This algorithm is very simple and its correctness is apparent. But its termination depends not only on the existence of a $\mathscr{J}$-complete tree but also on the choice of $\tilde{T}$, which has to be picked from the infinite set of trees with that special combinatorial structure.
It seems a natural idea to introduce a search level MAXLEVEL where the choice of $\tilde{T}$ can be revised. That is, if there exists a $\mathscr{J}$-complete tree with depth smaller or equal to MAXLEVEL, it can be determined by considering a finite number of trees $\tilde{T}$. If the depth-first search terminates for any of these trees successfully, that is,

[1] Recall that we assume a covered node to be combinatorially covered, which can be easily checked by comparison of index vector entries. The intention of introducing covered nodes was to obtain a finite structure for periodic infinite paths generated by $J_j \in \mathscr{J}$ such that combinatorially covered nodes are the essential ones.

any visited node in level MAXLEVEL is 1-bounded or covered, then $\hat{\rho}(\mathscr{A}) = 1$ is established due to Theorem 3.6.

Definition 4.1 *We call $\tilde{T} \subset T$ a* search tree of depth k *if*

1) *$\emptyset$ is root of $\tilde{T}$ and has m positive children*

2) *every node of length k is a leaf*

3) *every other node of $\tilde{T}$ has either exactly m positive children or exactly one negative child.*

To achieve an automated search by computer programs, we have to predefine the order in which the search trees of depth MAXLEVEL shall be considered. If we find a node K in level MAXLEVEL that is neither 1-bounded nor covered, the choice of search tree was not successful. Clearly, we do not want to re-start in root $\emptyset$ but rather retain the successfully visited branches unchanged and update the problematic part of the tree. That is, we update the search tree by modifying a subtree that starts in some prefix of K. We have to make sure that the updated tree still satisfies condition 3) of Definition 4.1 and want to retain as many successfully visited branches as possible. Therefore, the root should be chosen maximal such that it has a negative child. For any node K, we denote by

$$\text{NEGPARENT}(K) := \max\{R \in \mathscr{K} : \exists j < 0 \text{ s.t. } [R, j] \text{ is prefix of } K\}$$

the maximal prefix of K which has a negative child being a prefix of K as well, saying that NEGPARENT(K) exists if it is non-empty.

In the following, we identify a set-valued tree with its sets of nodes, writing $K \in \tilde{T}$ if K is a node of $\tilde{T}$. By *updating the subtree of* $P := \text{NEGPARENT}(K)$, we mean that $\tilde{T}$ is transformed by removing the nodes $[P,S] \in \tilde{T}$ and adding nodes $[P,S']$ such that the transformed tree $\tilde{T}$ remains a search tree of depth MAXLEVEL.
If $K = [P,i]$ is negative, then $\text{NEGPARENT}(K) = P$. In this case, the update is a replacement of the node K by either one of its negative siblings or by its m positive siblings.

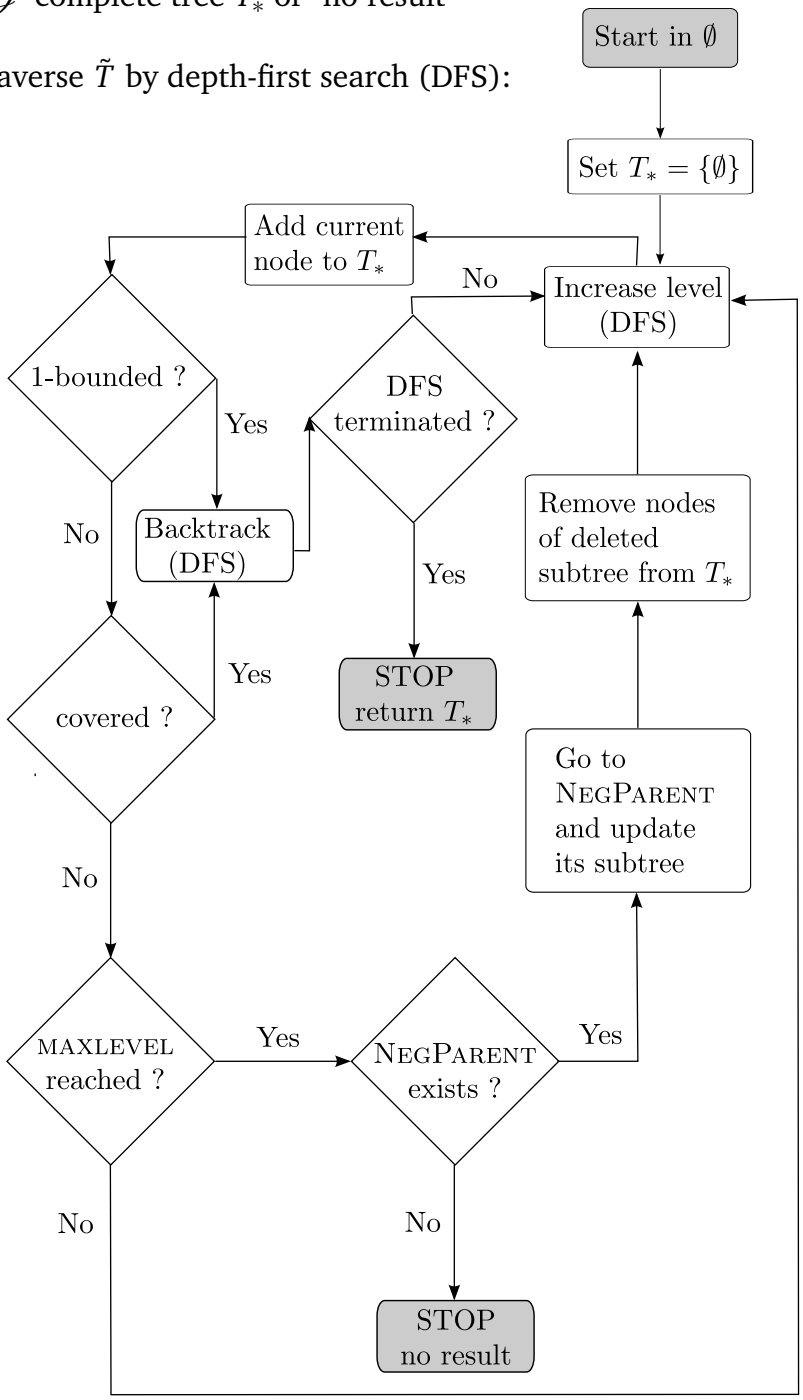
Clearly, there are different update strategies as well as different choices for the initial search tree. The basic concept for the algorithm is the following:

Algorithm 4.2 Basic concept: Search for $\mathscr{J}$-complete tree

Input: $\mathscr{A} = \{A_1, \ldots, A_m\}$
generator set $\mathscr{J} = \{J_1, \ldots, J_n\}$
submultiplicative matrix norm
maximal search level MAXLEVEL
initial search tree $\tilde{T}$ of depth MAXLEVEL

Output: $\mathscr{J}$-complete tree T_* or "no result"

Traverse $\tilde{T}$ by depth-first search (DFS):



If Algorithm 4.2 terminates with output "no result", no $\mathscr{J}$-complete tree with a depth up to MAXLEVEL was found. This could result from the fact that our initial guess for the FP-product was wrong but not necessarily: We cannot exclude that a $\mathscr{J}$-complete tree exists which could have been computed with increased MAXLEVEL or using a different update strategy.

An update strategy is *complete for* $\tilde{T}$ if it allows to generate any other search tree of depth MAXLEVEL from $\tilde{T}$. In that case, termination with output "no result" implies that no $\mathscr{J}$-complete tree with depth smaller or equal to MAXLEVEL exists.

For example, an update strategy which is complete for $\tilde{T} = \{[i,(-n)^k] : i \in \mathscr{I}_1, 0 \leq k \leq \text{MAXLEVEL} - 1\}$ is given by first trying all types of negative children in lexicographical order before passing on to positive children: Assume that a subtree with root R having the negative child $[R,j]$ is to be updated. The modified subtree is given by the set of nodes

$$\mathscr{K}_{\text{mod}} = \begin{cases} \bigcup_{i=1}^{m} \{[R,i,(-n)^k] : 0 \leq k \leq \text{MAXLEVEL} - |R| - 1\} & \text{if } j = -1 \\ \{[R,j+1,(-n)^k] : 0 \leq k \leq \text{MAXLEVEL} - |R| - 1\} & \text{else.} \end{cases}$$

4.2 Practical issues

In the worst-case, any complete update strategy has to deal with an enormous number of search trees that is exponentially increasing for raising MAXLEVEL. Not only for that reason it is sensible to choose an update strategy that selects just some of the search trees in a sophisticated way. There are search trees being very unlikely to have $\mathscr{J}$-complete subtrees. Consider for example $\tilde{T} = \{[i,j^k] : i \in \mathscr{I}_1, 0 \leq k \leq \text{MAXLEVEL} - 1\}$ for some $j < 0$. $\tilde{T}$ has no covered nodes, and if $[i,j^k]$ is 1-bounded for some $j < 0$, then $[i]$ is 1-bounded as well. Hence, $\tilde{T}$ has a $\mathscr{J}$-complete subtree if and only if $[i]$ is 1-bounded for any $i \in \mathscr{I}_1$. That is, the corresponding norm is extremal for $\mathscr{A}$.
Furthermore, if a search tree has no $\mathscr{J}$-complete subtree, we can conclude that a whole family of trees has none: Two succeeding negative nodes of the same type $[P,j]$ and $[P,j,j]$, $j < 0$ code the same set of matrix products. For $S \in \mathscr{K}_{\text{MAXLEVEL}-|P|-1}$, denote by S_k its prefix of length k. If the path $([P,j,S_k])_{1 \leq k \leq |S|}$ contains neither a 1-bounded nor a covered node, the same holds for $([P,j^\ell,S_k])_{1 \leq k \leq |S|-\ell+1}$. The update strategy should therefore avoid to generate a tree which contains two or more succeeding negative nodes of the same type.

On one hand, if $\hat{\rho}(\mathscr{A}) = 1$ is true and the chosen norm does not satisfy[2] $\|A_J\| = 1$, a necessary condition for a search tree $\tilde{T}$ to have a $\mathscr{J}$-complete subtree is the existence of at least one negative node in $\tilde{T}$. On the other hand, the computational costs for checking if a node is 1-bounded increase with the number of negative

[2] $\|A_J\| = 1$ holds in particular if $\|\cdot\|$ is an extremal norm, see [GWZ05].

prefixes, see also Section 4.3. Introducing a parameter NEGLIMIT, the update strategy therefore should control that the number of negative nodes in any path of the generated search tree does not exceed NEGLIMIT.
A $\mathscr{J}$-complete tree is not unique, so that different values of NEGLIMIT may lead to different trees, see the example in Section 7.6. Choosing NEGLIMIT too large therefore might lead to unnecessary computations. But there are situations with two or even more nested infinite paths, where choosing NEGLIMIT > 1 is mandatory. This is the case for the family $\mathscr{C}$ in Section 7.1.

From a heuristic point of view, a search tree is more or less promising depending on the location of negative nodes. For $j < 0$ and $J := J_{-j}$ and arbitrary $\ell \in \mathbb{N}_0$, $P, S \in \mathscr{K}$, we expect rather $[P, J^\ell, j, S]$ to be 1-bounded than $[P, j, S]$. To motivate this, assume that $(A_J^k)_k$ converges to a limit matrix A_J^∞. If $\|A_I A_J^\infty A_P\| < 1$ for some $I \in \mathscr{I}$, then there exists $\ell \in \mathbb{N}_0$ such that $\|A_I A_J^k A_P\| \le 1$ for $k \ge \ell$, implying that $[P, J^\ell, j, I]$ is 1-bounded. In contrast, 1-boundedness of $[P, j, I]$ means that $\|A_I A_J^k A_P\| \le 1$ for all $k \in \mathbb{N}_0$, a much stronger requirement, which, if satisfied, implies also 1-boundedness of $[P, J^\ell, j, S]$. Therefore, the implemented update strategy has an input parameter STARTLEVEL and involves only search trees whose negative nodes are of the form $[P, J_{-j}^\ell, j]$ for $\ell \ge \lfloor (\text{STARTLEVEL} - 1)/|J_{-j}| \rfloor$.

The computations for completely positive nodes are very efficient in comparison to negative nodes or their successors. Therefore, we do not want to compute a long path with a negative node $[P, j]$ in a low level. Instead, we prefer to stop the computation even before reaching MAXLEVEL and to update the subtree such that $[P, J^\ell, j]$ is node of the search tree. The parameter AFTERNEG, by default set to $\lceil \text{MAXLEVEL}/3 \rceil$, bounds the length of a node which has a negative prefix. For sake of simplicity, this parameter will be neglected in further descriptions of the algorithm.

Saving a search tree $\tilde{T}$ of a potentially high depth that is constantly updated requires lots of RAM. In fact, it is not necessary to know all successors of the current node, it suffices to know its children. Instead of traversing a search tree that is a priori fixed by input or update strategy, the search strategy should work locally. The implemented algorithm corresponds in principle to Algorithm 4.2 but uses, whenever the level is to be increased, a decision routine DECISION whose output specifies the children of the current node.

Since the decision routine operates locally, we have to define the backtracking appropriately to make sure that the combinatorial structure of a $\mathscr{J}$-complete tree is achieved. If a positive node $[P, 1]$ was analyzed, the algorithm should not return T_* before its $m - 1$ positive siblings were visited. If K is a node with an entry $1 \le i \le m - 1$, we define

$$\text{UNFINISHED}(K) := \max\{[P, i+1] : [P, i] \text{ prefix of } K, 1 \le i \le m-1\}.$$

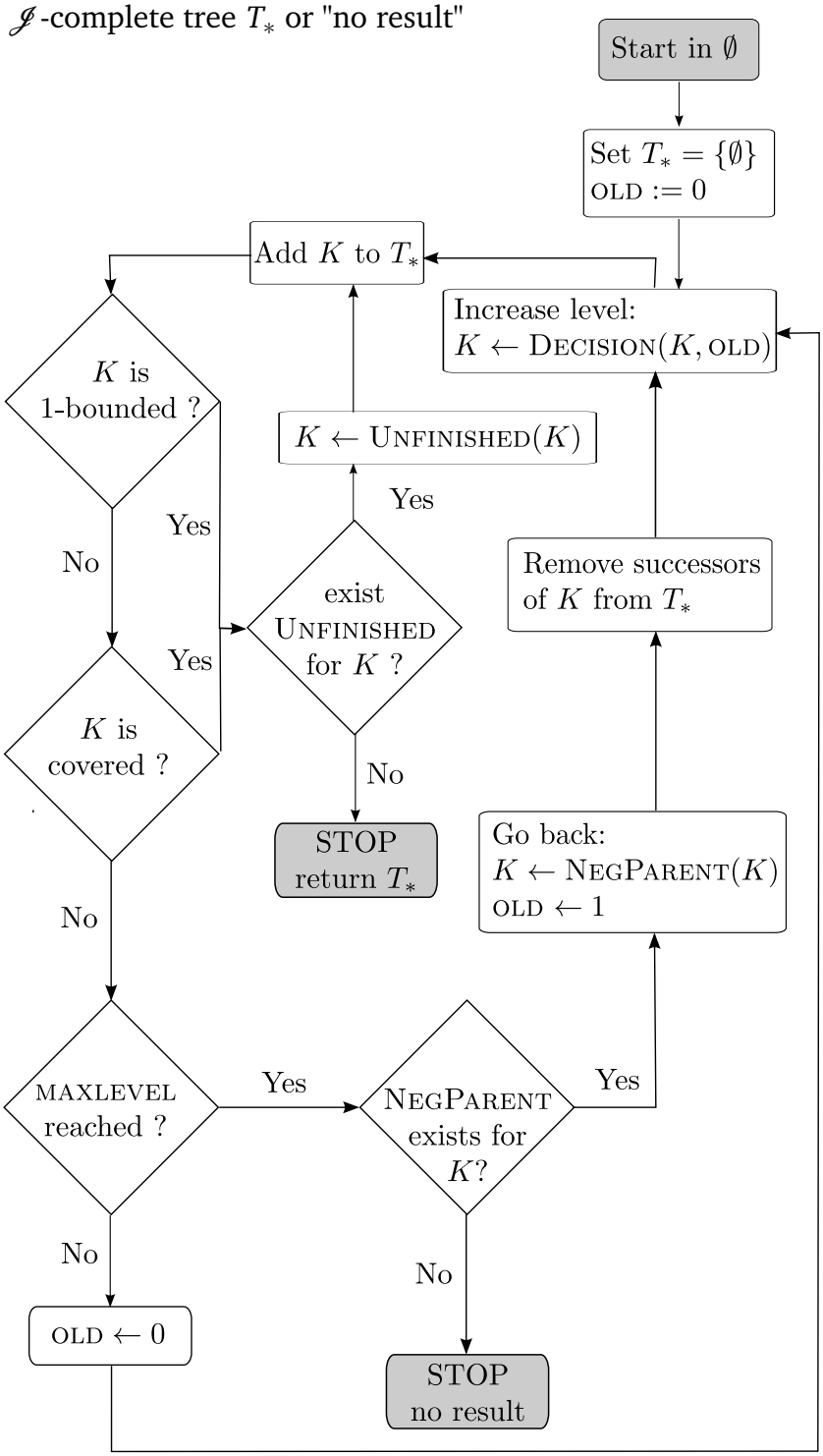
When the backtrack criterion is satisfied, the algorithm has to check if UNFINISHED(K) exists for the current node K.

This leads to the following algorithm:

Algorithm 4.3 Search for $\mathscr{J}$-complete tree

Input: $\mathscr{A} = \{A_1, \ldots, A_m\}$
generator set $\mathscr{J} = \{J_1, \ldots, J_n\}$
submultiplicative matrix norm
maximal search level MAXLEVEL

Output: $\mathscr{J}$-complete tree T_* or "no result"



The implemented decision routine works as follows.

Algorithm 4.4 Decision routine

Input: node K
generator set $\mathscr{J} = \{J_1, \ldots, J_n\}$
flag argument OLD
parameters MAXLEVEL, STARTLEVEL, NEGLIMIT

Output: Child $[K, i]$ with $i \in \{-n, \ldots, -1\} \cup \{1\}$

$i \leftarrow 1$
if OLD == 1 do
  return $[K, i]$
else
  NEGNUMB $\leftarrow$ number of negative entries in K
  if $|K| + 1 \geq$ STARTLEVEL && $|K| + 1 <$ MAXLEVEL && NEGNUMB $<$ NEGLIMIT
    for j from 1 to n do
      if $J_i^{\lfloor (\text{STARTLEVEL}-1)/|J_i| \rfloor}$ is suffix of K
        $i \leftarrow -j$
        break
      end
    end
  end
  return $[K, i]$
end

The backtrack criterion is satisfied if the current node K is either covered or 1-bounded. The check if a node is covered corresponds to comparing the entries of $[-j, J_j]$ and the suffix of K with the according length. The more complicated and computationally expensive check is the one for 1-boundedness. If $K \in \mathscr{I}$, we simply have to compute the norm of one matrix product. But if $K \notin \mathscr{I}$, $\mathscr{A}_K$ is an infinite set of products and K is 1-bounded if and only if $\|\mathscr{A}_K\| = \sup\{\|A\| : A \in \mathscr{A}_K\} \leq 1$. In practice, we determine instead of $\|\mathscr{A}_K\|$ an upper bound N_K as discussed in Section 4.3. Additionally, when computing N_K numerically, we accept K to be 1-bounded only if $N_K < 1 - \text{SAFETYCONST}$ to avoid incorrectness due to numerical inaccuracy. The same holds for $K \in \mathscr{I}$. The examples in Chapter 7 are typically computed with $\text{SAFETYCONST} = 10^{-7}$. For these reasons, a 1-bounded node might not be detected, especially if it is not strictly 1-bounded.

4.3 Upper bounds for norms

When implementing Algorithm 4.3, it is to decide how to compute an upper bound of the norm $\|\mathscr{A}_K\|$ for $K \notin \mathscr{I}$. A tight upper bound for a nodes' norm is of great importance since 1-boundedness is a backtrack criterion. Equally, the runtime of the algorithm depends strongly on the computation of norms since it has to be performed in every node. In order to achieve numerical efficiency as well as a tight bound, the structure of the coded matrices should be taken into account.

Two approaches to a computation of an upper bound N_K, having different requirements, are presented in the following. The approach presented in Section 4.3.1 requires that, for any $J \in \mathscr{J}$, A_J has a unique real leading eigenvalue with equal algebraic and geometric multiplicity. It then benefits from the convergence of powers A_J^k and leads in principle to arbitrarily close bounds. In contrast to that, the approach introduced in Section 4.3.2 requires that complex eigenvalues of matrix products only occur as conjugate pairs, as it is the case for a real matrix family $\mathscr{A}$. In general, this approach is numerically expensive for increasing dimension such that most of the examples in Chapter 7 are computed using the first approach. Both approaches assume the generator matrices A_J for any $J \in \mathscr{J}$ to be diagonalizable. As the Jordan normal form is sensitive to perturbations, it is numerically unstable. Therefore, a generator matrix which is not diagonalizable is hardly encountered in practice.

4.3.1 Balls of matrices

In the following, we assume that for any $J \in \mathscr{J}$, there is a unique leading eigenvalue of A_J being real and having equal algebraic and geometric multiplicity. In case that J is a strong generator, we request the leading eigenvalue additionally to be positive[3]. To simplify forthcoming arguments, we assume that A_J is diagonalizable for any $J \in \mathscr{J}$. The matrices described by a node K in some sense differ only by powers of generator matrices. When those powers tend to infinity, the matrices converge. We use this property to define a superset of $\mathscr{A}_K$ which provides an upper bound for $\|\mathscr{A}\|_K$ that is easy to determine.
For any $j \in \{1, \ldots, n\}$ and $J := J_j$, the powers A_J^k converge to a limit matrix A_J^∞. Hence, there exists a ball on $\mathbb{C}^{d\times d}$ with midpoint A_J^∞ which contains $\mathscr{A}_{[-j]}$. To obtain a superset for $\mathscr{A}_K$, $K \in \mathscr{K}$, we appropriately define a multiplication on the space of matrix balls. Furthermore, by defining the extent $\Xi(\cdot)$ of a ball such that $\|\mathscr{A}_K\| \leq \Xi(B)$ if $\mathscr{A}_K \subseteq B$, the computation of an upper bound bases on determining an enclosing ball for $\mathscr{A}_K$.

[3] If J is the only generator, this can always be achieved by appropriate scaling of the family. In principle, the approach applies also in case of a negative leading eigenvalue. Then, $(A_J^k)_k$ has two convergent subsequences and a superset is given by the union of two matrix balls centered in the two limit points.

Definition 4.5 *Given a norm $\|\cdot\|$, we denote by $B = (C, r)$ with $C \in \mathbb{C}^{d\times d}$ and $r \in \mathbb{R}_{\geq 0}$ the matrix ball $B := \{X \in \mathbb{C}^{d\times d} : \|C - X\| \leq r\}$. Then, we define a* product of balls *$B = (C, r)$ and $\tilde{B} = \left(\tilde{C}, \tilde{r}\right)$ by*

$$B * \tilde{B} := \left(C \cdot \tilde{C}, \|C\| \cdot \tilde{r} + \|\tilde{C}\| \cdot r + r \cdot \tilde{r}\right).$$

Further, we define the extent of a ball *$B = (C, r)$ by*

$$\Xi(B) := \|C\| + r.$$

Obviously, $\|X\| \leq \Xi(B)$ for all $X \in B$. For any finite number of sets, the product of their enclosing balls is a ball enclosing their product. This results by induction from the following lemma:

Lemma 4.6 *For $\mathscr{A}_S \subseteq B$ and $\mathscr{A}_P \subseteq \tilde{B}$, it holds that $\mathscr{A}_{[P,S]} \subseteq B * \tilde{B}$.*

Proof. Let $B = (C, r)$ and $\tilde{B} = (\tilde{C}, \tilde{r})$. An element of $\mathscr{A}_{[P,S]}$ is a product $A \cdot \tilde{A}$ with $A \in \mathscr{A}_S$ and $\tilde{A} \in \mathscr{A}_P$. By assumption, $A \in B$ and $\tilde{A} \in \tilde{B}$.

$$\begin{aligned}\|C \cdot \tilde{C} - A \cdot \tilde{A}\| &= \|C \cdot (\tilde{C} - \tilde{A}) + (C - A) \cdot \tilde{A}\| \\ &\leq \|C\| \cdot \|\tilde{C} - \tilde{A}\| + \|C - A\| \cdot (\|\tilde{C}\| + \tilde{r}) \\ &\leq \|C\| \cdot \tilde{r} + \|\tilde{C}\| \cdot r + r \cdot \tilde{r}.\end{aligned}$$

Therewith, $A \cdot \tilde{A} \in B * \tilde{B}$. □

Consider a node $K \in \mathscr{K}$. Then $\mathscr{A}_K$ is a finite product with factors $\mathscr{A}_I$, $I \in \mathscr{I}$ and/or $\mathscr{A}_{[-j]}$, $j \in \{1, \ldots, n\}$. In case of $I \in \mathscr{I}$, it is trivial that $\mathscr{A}_I = \{A_I\} \in (A_I, 0)$. It remains to find an enclosing ball for $\mathscr{A}_{[-j]}$. By assumption, A_J with $J := J_j$ is diagonalizable, $\rho(A_J) \leq 1$ and all eigenvalues with absolute value 1 are real and positive. So there is $0 \leq \ell \leq d$ such that w.l.o.g. $\lambda_i = 1$ for $1 \leq i \leq \ell$ and $|\lambda_i| < 1$ for $\ell + 1 \leq i \leq d$. Then, with u_i^T and v_i being the left resp. right eigenvector corresponding to λ_i, the matrices $T_i = \frac{v_i \cdot u_i^T}{u_i^T \cdot v_i}$ allow the representation

$$A_J^k = \sum_{i=1}^{\ell} T_i + \sum_{i=\ell+1}^{d} \lambda_i^k T_i. \tag{4.1}$$

Obviously, $A_J^\infty := \lim_{k\to\infty} A_J^k = \sum_{i=1}^{\ell} T_i$. Defining

$$\begin{aligned} r_J &: \mathbb{N} \to \mathbb{R}_{\geq 0} \\ r_J(k) &= \sum_{i=\ell+1}^{d} |\lambda_i|^k \|T_i\|, \end{aligned}$$

r_J is a monotonically decreasing function with $\lim_{k\to\infty} r_J(k) = 0$ such that

$$A_J^k \in \left(A_J^\infty, r_J(\tilde{k})\right) \tag{4.2}$$

holds for all $k \geq \tilde{k}$. The following result is now a direct consequence of Lemma 4.6.

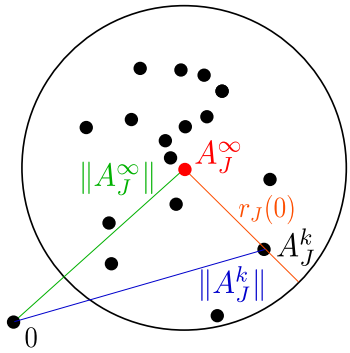


Figure 4.1: Visualization of a ball $\left(A_J^\infty, r_J(0)\right)$. The matrix powers A^k converging to A_J^∞ are indicated for $0 \leq k \leq 16$ by the black dots. 0 denotes the zero matrix.

Corollary 4.7 *Consider an arbitrary node $K \in \mathscr{K}$. Let $h \in \mathbb{N}$ be the number of negative entries, then $K = [I_1, -i_1, \ldots, I_h, -i_h, I_{h+1}]$ for some $I_1, \ldots, I_{h+1} \in \mathscr{I}$ and $i_1, \ldots, i_h \in \{1, \ldots, n\}$. With*

$$B_K := \left(A_{I_{h+1}}, 0\right) * \left(A_{J_{i_h}}^\infty, r_{J_{i_h}}(0)\right) * \left(A_{I_h}, 0\right) * \cdots * \left(A_{J_{i_1}}^\infty, r_{J_{i_1}}(0)\right) * \left(A_{I_1}, 0\right),$$

it holds that

$$\mathscr{A}_K \subseteq B_K$$

and therewith $\|\mathscr{A}_K\| \leq \Xi(B_K)$.

An advantage of this approach is that the balls $\left(A_J^\infty, r_J(0)\right)$ can be computed a priori for each $J \in \mathscr{J}$. In each node, the computation of the upper bound then reduces to a multiplication of balls and determination of the extent of this product.

When determining a superset of $\mathscr{A}_K$, the ball B_K is one possibility. But also a union of balls can be considered:

Definition 4.8 *Let $\mathbf{B} = [B_1, \ldots, B_r], r \in \mathbb{N}$ be a vector of balls. We say that $A \in \mathbf{B}$ if $A \in \bigcup_{i=1}^r B_i$ and define $\Xi(\mathbf{B}) := \max_{i \in \{1,\ldots,r\}} \Xi(B_i)$. Furthermore, we extend the multiplication of balls to vectors of balls: For $\mathbf{B}$ and $\tilde{\mathbf{B}} = [\tilde{B}_1, \ldots, \tilde{B}_{\tilde{r}}]$, $\mathbf{B} * \tilde{\mathbf{B}}$ is understood as a vector of balls of length $r \cdot \tilde{r}$ with entries $B_i * \tilde{B}_j$, $i \in \{1, \ldots, r\}, j \in \{1, \ldots, \tilde{r}\}$.*

It is not difficult to see that $\|A\| \leq \Xi(\mathbf{B})$ for $A \in \mathbf{B}$, and that $A \cdot \tilde{A} \in \mathbf{B} * \tilde{\mathbf{B}}$ for $A \in \mathbf{B}$ and $\tilde{A} \in \tilde{\mathbf{B}}$. Most important, since we aim to determine a tight bound for $\mathscr{A}_K$, the extent of $\mathbf{B} = [B_1, \ldots, B_r]$ is smaller than or at most equal to the extent of any ball enclosing $\bigcup_{i=1}^r B_i$.

For $k_0 \in \mathbb{N}$, let

$$\mathbf{A}_{J,k_0} := \left[(A_J^0, 0), \ldots, (A_J^{k_0-1}, 0), \left(A_J^\infty, r_J(k_0)\right) \right].$$

With (4.2), it follows that

$$\mathscr{A}_{[-j]} \subseteq \mathbf{A}_{J,k_0}. \tag{4.3}$$

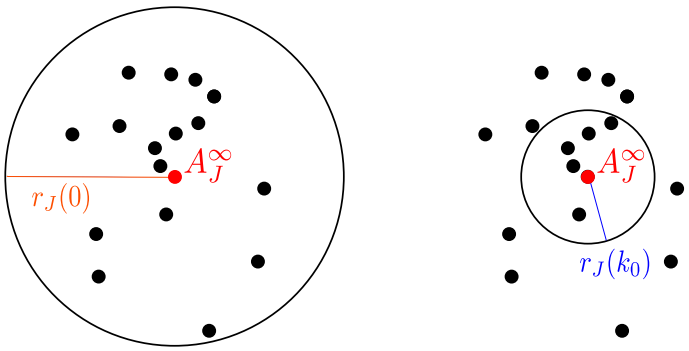


Figure 4.2: Supersets of $\{A_J^k : k \in \mathbb{N}_0\}$:
An enclosing ball is given by $\left(A_J^\infty, r_J(0)\right)$ (*left*). The union of $\left(A_J^\infty, r_J(k_0)\right)$ and the non-contained trivial balls (*right*) is described by the ball vector $\mathbf{A}_{J,k_0}$ having length 12.

The following observation is not important for the theory, but useful in practice to keep ball vectors short: If A_J^i is contained in $\left(A_J^\infty, r_J(k_0)\right)$, then $\mathbf{A}_{J,k_0}$ can be reduced to $\left[(A_J^0, 0), \ldots, (A_J^{i-1}, 0), (A_J^{i+1}, 0) \ldots, (A_J^{k_0-1}, 0) \left(A_J^\infty, r_J(k_0)\right) \right]$. The union of entries remains the same set, and the extent does not change.
An analogous result to Corollary 4.7 with a closer upper bound follows immediately with (4.3):

Corollary 4.9 *For K as in Corollary 4.7, $k_{0,j} \in \mathbb{N}_0$ for $j = 1, \ldots, n$, and*

$$\mathbf{B}_\mathbf{K} := \left(\mathbf{A}_{\mathbf{I}_{h+1}}, \mathbf{0}\right) * \mathbf{A}_{\mathbf{J}_{i_h}, \mathbf{k}_{0,i_h}} * \left(\mathbf{A}_{\mathbf{I}_h}, \mathbf{0}\right) * \cdots * \mathbf{A}_{\mathbf{J}_{i_1}, \mathbf{k}_{0,i_1}} * \left(\mathbf{A}_{\mathbf{I}_1}, \mathbf{0}\right),$$

it holds that

$$\mathscr{A}_K \subseteq \mathbf{B}_\mathbf{K}$$

and therewith $\|\mathscr{A}_K\| \leq \Xi(\mathbf{B}_\mathbf{K})$.

$\mathbf{A}_{J,k_0}$ can be computed and reduced a priori. In a node, the determination of the bound reduces to computing the ball products, determining the extent of each entry, that is, adding the midpoints' norm to the radius, and eventually finding the maximal extent.

It remains the question of how to choose the constants $k_{0,j}$ appropriately. Details are postponed to Section 4.3.3 in which the implemented solution is described. There is a tradeoff since the upper bound becomes tight for large $k_{0,j}$ but the vector of balls $\mathbf{A}_{J_j,k_{0,j}}$ becomes long such that the runtime in each node increases. The implemented solution bases on the following idea. If the ball $\left(A_J^\infty, r_J(k_0)\right)$ with $J := J_j$ is contained in a ball (C,r), then

$$\mathscr{A}_{[-j]} \subseteq \bigcup_{0\le\ell<k_0} \{A_J^\ell\} \cup (C,r) \tag{4.4}$$

holds. Therefore, we can substitute $\left(A_J^\infty, r_J(k_0)\right)$ in the ball vector $\mathbf{A}_{J,k_0}$ by (C,r). If C is chosen such that A_J^k is contained in (C,r) for $k < k_0$, we can reduce $\mathbf{A}_{J,k_0}$ by these powers A_J^k and obtain a valid upper bound for a shorter vector of balls, see Figure 4.3 for an illustration. The tightness of the bound then depends on r. Details on how to find (C,r) are postponed to Section 4.3.3.

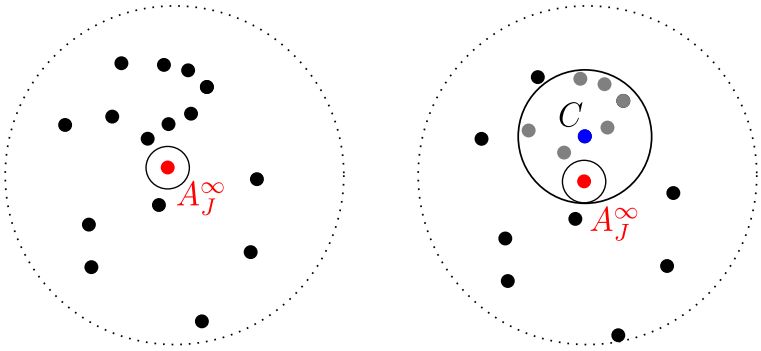


Figure 4.3: The figure on the left visualizes the powers A_J^k that are not contained in $\left(A_J^\infty, r_J(k_0)\right)$. Choosing (B,r) as displayed on the right, the number is significantly reduced.

The problem of long ball vectors is aggravated in case of weak generators: To be useful in practice, a weak generator J has a spectral radius $\rho(A_J) = 1-\varepsilon$. As a consequence, convergence of the powers to the zero matrix is very slow, and so is the decay of $r_J(k)$. To obtain a tight bound for $\mathscr{A}_K$, k_0 then has to be chosen very large, leading to long ball vectors $\mathbf{A}_{J,k_0}$ although the subdominant eigenvalue might converge rapidly to zero. To avoid this problem, we change the midpoint of the non-trivial ball by separating the leading eigenvalue: Again, we can assume w.l.o.g. that $\lambda_1 = \cdots = \lambda_\ell$, for some $1 \le \ell \le d$, $|\lambda_1| = 1-\varepsilon$ and $|\lambda_i| < |\lambda_1|$ for $\ell+1 \le i \le d$. Since

$$\left\|A_J^k\right\| = \left\|\sum_{i=1}^d \lambda_i^k T_i\right\| \le \left\|\sum_{i=1}^\ell T_i\right\| + \left\|\sum_{i=\ell+1}^d \lambda_i^k T_i\right\|, \tag{4.5}$$

all the arguments from above hold equally when substituting the zero matrix A_J^∞ by $\tilde{A}_J := \sum_{i=1}^\ell T_i$. We could think of $\tilde{A}_J$ as the limit matrix when changing for A_J the leading eigenvalue λ with $|\lambda| = 1-\varepsilon$ to $\lambda = 1$.

4.3.2 Bounding the eigenvalues

The following approach leads to an alternative upper bound N_K for $K \notin \mathscr{I}$. The basic idea is to substitute the powers of eigenvalues by combinations of upper and lower bounds and therewith to get an upper bound for all matrix powers coded by the node. In the following, we assume that all complex eigenvalues of A_J, $J \in \mathscr{J}$ occur as conjugate pairs.

Consider a node K with ℓ negative entries. Then K is of the form

$$K = \left[I_1, -j_1, I_2, \ldots, -j_\ell, I_{\ell+1}\right]$$

with $j_1, \ldots, j_\ell \in \{1, \ldots, n\}$ and $I_i \in \mathscr{I}$ for $i = 1, \ldots, \ell+1$. Hence,

$$\mathscr{A}_K = \{A_{I_{\ell+1}} A_{J_{j_\ell}}^{k_\ell} A_{I_\ell} \cdots A_{J_{j_1}}^{k_1} A_{I_1} : k_1, \ldots, k_\ell \in \mathbb{N}_0\}$$

and

$$\|A_K\| = \sup_{k_1,\ldots,k_\ell \in \mathbb{N}_0} \|A_{I_{\ell+1}} A_{J_{j_\ell}}^{k_\ell} A_{I_\ell} \cdots A_{J_{j_1}}^{k_1} A_{I_1}\|. \tag{4.6}$$

In the following, we abbreviate $P := I_1$, $S := I_2$, $J := J_{j_1}$ $k := k_1$ $j := j_1$ and $s := s_1$ whenever it is convenient.
We assumed A_{J_j} to be diagonalizable for all $j \in \{1, \ldots, n\}$, denoting

$$A_{J_j} = V_j D_j U_j$$

with $U_j = V_j^{-1}$ and $D_j = \operatorname{diag}\left(\lambda_{j,1}, \lambda_{j,2}, \ldots, \lambda_{j,d}\right)$. We denote by $M_{i,:}$ the i-th row and by $M_{:,j}$ the j-th column of a matrix M. Then, with $H_s := (A_S V_j)_{:,s} \cdot (U_j A_P)_{s,:}$, it is

$$A_S A_J^k A_P = A_S V_j D_j U_j A_P = \sum_{s=1}^{d} \lambda_{j,s}^k \cdot H_s.$$

To avoid powers of complex numbers, we transform as follows:
If $\lambda_{j,s}$ is real, we define

$$T_s := H_s \qquad \text{and} \qquad \mu_{j,s,k} := \lambda_{j,s}^k.$$

Otherwise, there is by assumption $\tilde{s}$ such that $\lambda_{j,s}$ and $\lambda_{j,\tilde{s}}$ are complex conjugates. Then we define

$$T_s := H_s + H_{\tilde{s}}, \qquad \text{and} \qquad T_{\tilde{s}} := i \cdot (H_s - H_{\tilde{s}})$$

together with

$$\mu_{j,s,k} := \Re(\lambda_{j,s}^k) \qquad \text{and} \qquad \mu_{j,\tilde{s},k} := \Im(\lambda_{j,s}^k).$$

Due to

$$\lambda_{j,s}^k H_s + \lambda_{j,\tilde{s}}^k H_{\tilde{s}} = \Re(\lambda_{j,s}^k) \cdot (H_s + H_{\tilde{s}}) + \Im(\lambda_{j,s}^k) \cdot i(H_s - H_{\tilde{s}}),$$

it is

$$\sum_{s=1}^{d} \mu_{j,s,k} \cdot T_s = \sum_{s=1}^{d} \lambda_{j,s}^k \cdot H_s,$$

and the scalars $\mu_{j,s,k}$ are real.
Substituting $A_S A_J^k A_P = \sum_{s=1}^{d} \mu_{j,s,k} \cdot T_s$ into (4.6) and repeating the arguments for $i = 2, \ldots, \ell$ leads to defining μ_{j_i,s_i,k_i} analogously to $\mu_{j,s,k}$, and to the recursive[4] definition

$$H_{(s_1,\ldots,s_i)} = (A_{I_{i+1}} V_{j_i})_{:,s_i} \cdot (U_{j_i} T_{(s_1,\ldots,s_{i-1})})_{s_i,:}$$

Depending on whether λ_{j_i,s_i} is real or complex with conjugate $\lambda_{j_i,\tilde{s}_i}$, either

$$T_{(s_1,\ldots,s_i)} := H_{(s_1,\ldots,s_i)}$$

or

$$T_{(s_1,\ldots,s_i)} := H_{(s_1,\ldots,s_i)} + H_{(\tilde{s}_1,\ldots,\tilde{s}_i)} \qquad \text{and} \qquad T_{(\tilde{s}_1,\ldots,\tilde{s}_i)} := i \cdot \left(H_{(s_1,\ldots,s_i)} - H_{(\tilde{s}_1,\ldots,\tilde{s}_i)} \right).$$

This leads to

$$\|A_K\| = \sup_{k_1,\ldots,k_\ell} \left\| \sum_{s_1,\ldots,s_\ell=1}^{d} \mu_{j_1,s_1,k_1} \cdots \mu_{j_\ell,s_\ell,k_\ell} \cdot T_{(s_1,\ldots,s_\ell)} \right\|. \tag{4.7}$$

Since $|\lambda| \leq 1$, $\Re(\lambda^k)$ and $\Im(\lambda^k)$ are bounded. With $\lambda = re^{i\varphi}$, it is

$$\inf_{k\in\mathbb{N}_0} \{r^k \cos(k\varphi)\} \leq \Re(\lambda^k) \leq \sup_{k\in\mathbb{N}_0} \{r^k \cos(k\varphi)\} \tag{4.8}$$

and

$$\inf_{k\in\mathbb{N}_0} \{r^k \sin(k\varphi)\} \leq \Im(\lambda^k) \leq \sup_{k\in\mathbb{N}_0} \{r^k \sin(k\varphi)\}. \tag{4.9}$$

Figure 4.4 illustrates these bounds. Real eigenvalues are a particular case and do not need to be considered separately. The bounding box of λ^k then reduces to a line or, if $\lambda = 1$, to a point.

[4] The definition is recursive since $T_{(s_1,\ldots,s_{i-1})}$ depends on $H_{(s_1,\ldots,s_{i-1})}$.

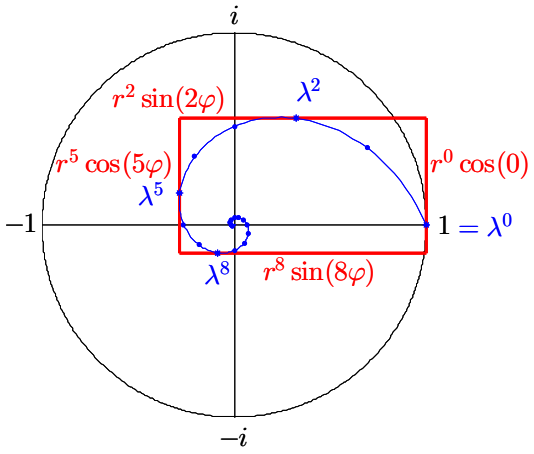


Figure 4.4: Bounding box for λ^k, exemplarily with $\lambda = 0.8e^{i\frac{\pi}{6}}$.

Therewith, upper and lower bounds $a_{s_i} \leq \mu_{j_i,s_i,k_i} \leq b_{s_i}$ are given in case $\mu_{j_i,s_i,k} = \Re(\lambda^k_{j_i,s_i})$ by (4.8) or in case $\mu_{j_i,s_i,k} = \Im(\lambda^k_{j_i,\tilde{s}_i})$ by (4.9) for any $i \in \{1,\dots,\ell\}$ and $s_i \in \{1,\dots,d\}$. Set $\Omega_{j_i} := [a_1,b_1]\times[a_2,b_2]\times\cdots\times[a_d,b_d]$ and $\Omega := \Omega_{j_1}\times\cdots\times\Omega_{j_\ell}$. For any power k, the vector $(\mu_{j_i,1,k},\mu_{j_i,2,k},\dots,\mu_{j_i,d,k})$ is an element of Ω_{j_i}. Using this fact, we define a function f whose maximum value bounds $\mathscr{A}_K$.

Proposition 4.10 *The maximum of the function $f : \Omega \to \mathbb{R}_+$,*

$$f(x_1,\dots,x_\ell) = \left\| \sum_{s_1,\dots,s_\ell=1}^{d} (x_1)_{s_1}\dots(x_\ell)_{s_\ell}\cdot T_{(s_1,\dots,s_\ell)} \right\|$$

is attained at a vertex of Ω, and is an upper bound of $\|\mathscr{A}_K\|$.

Proof. Recalling (4.7) and the definition of Ω, it is not difficult to see that $\|\mathscr{A}_K\|$ is bounded by the maximal value of f.
Obviously, Ω is convex and bounded for any $\ell \in \mathbb{N}$. We show by induction on ℓ that the maximum is attained at a vertex. Set $x := x_1$. For $\ell = 1$,

$$f(x) = \left\| \sum_{s=1}^{d} x_s T_s \right\|$$

is convex, implying the claim. In the induction step, we show that $f(x_1,\dots,x_{\ell+1})$ attains its maximum at a vertex of $\Omega = \Omega_{j_1}\times\dots\Omega_{j_{\ell+1}}$. Consider fixed $(\tilde{x}_1,\dots,\tilde{x}_{\ell+1}) \in \Omega$. Define

$$f_{\tilde{x}_{\ell+1}}(x_1,\dots,x_\ell) := f(x_1,\dots,x_\ell,\tilde{x}_{\ell+1})$$

and

$$f_{(\tilde{x}_1,\dots,\tilde{x}_\ell)}(x_{\ell+1}) := f(\tilde{x}_1,\dots,\tilde{x}_\ell,x_{\ell+1}).$$

By the induction hypothesis, $f_{\tilde{x}_{\ell+1}}$ attains the maximal value at some vertex $(v_1,\dots,v_\ell)$ of $\Omega_{j_1}\times\dots\Omega_{j_\ell}$, and

$$f(v_1,\dots,v_\ell,\tilde{x}_{\ell+1}) = f_{\tilde{x}_{\ell+1}}(v_1,\dots,v_\ell) \geq f_{\tilde{x}_{\ell+1}}(\tilde{x}_1,\dots,\tilde{x}_\ell) = f(\tilde{x}_1,\dots,\tilde{x}_{\ell+1}).$$

Furthermore, $f_{(v_1,\dots,v_\ell)}$ attains the maximal value at a vertex w of $\Omega_{j_{\ell+1}}$. Hence,

$$f(v_1,\dots,v_\ell,w) = f_{(v_1,\dots,v_\ell)}(w) \geq f_{(v_1,\dots,v_\ell)}(\tilde{x}_{\ell+1}) = f_{\tilde{x}_{\ell+1}}(v_1,\dots,v_\ell).$$

Altogether, $(v_1,\dots,v_\ell,w)$ is a vertex of Ω satisfying

$$f(v_1,\dots,v_\ell,w) \geq f(\tilde{x}_1,\dots,\tilde{x}_{\ell+1})$$

for arbitrarily chosen $(\tilde{x}_1,\dots,\tilde{x}_{\ell+1})\in\Omega$. □

Remark: Due to Proposition 4.10, an algorithm which searches all vertices of Ω to find the maximum value will lead to the desired upper bound for the node K. Now, there are about[5] 2^{d^ℓ} vertices, with d being the space dimension and ℓ the number of negative entries in K. Hence, this is numerically expensive already for relatively small values of ℓ and d. Though Ω_{j_i} can be computed a priori for each generator J_{j_i}, $j_i\in\{1,\dots,n\}$, the matrices $T_{(s_1,\dots,s_\ell)}$ depend on K.

If $\ell=1$, the bound for $\mathscr{A}_K$ can be improved by computing the norm of the first powers separately such that the upper bound is to be determined for eigenvalue powers starting from some $k_0\in\mathbb{N}$ which allows smaller bounding boxes,

$$\begin{aligned}\|\mathscr{A}_K\| &= \sup_{k\geq 0} A_S A_J^k A_P = \sup_{k\geq 0}\sum_{s=1}^{d}\lambda_{j,s,k}\cdot T_s \\ &= \max\left\{\max_{k\leq k_0-1}\|A_S A_J^k A_P\|, \sup_{k\geq 0}\sum_{s=1}^{d}\mu_{j,s,k_0+k}\cdot T_s\right\}.\end{aligned}$$

For $\lambda = re^{i\varphi}$, bounds are given by

$$\inf_{k\geq k_0}\{r^k\cos(\varphi k)\} \leq \Re(\lambda^{k0+k}) \leq \sup_{k\geq k_0}\{r^k\cos(\varphi k)\}$$

and

$$\inf_{k\geq k_0}\{r^k\sin(\varphi k)\} \leq \Im(\lambda^{k0+k}) \leq \sup_{k\geq k_0}\{r^k\sin(\varphi k)\}.$$

Using these bounds to define Ω, Proposition 4.10 can be adapted. Figure 4.5 visualizes the bounding box exemplarily for $k_0=4$ in comparison with the original bounding box.

[5] To be precise, eigenvalues $\lambda=1$ decrease the number of vertices since lower and upper bounds coincide.

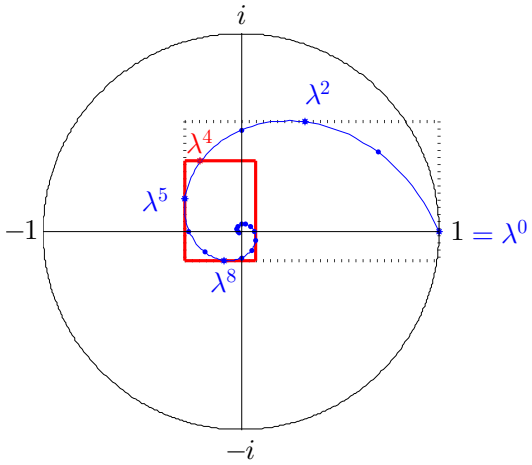


Figure 4.5: Bounding box for λ^k with $k \geq k_0 = 4$ (solid line) and for comparison $k \geq 0$ (dotted line). In this example, $\lambda = 0.8e^{i\frac{\pi}{6}}$.

Choosing $k_0 > 0$ improves the check for 1-boundedness even for small k_0. Additionally, if evaluating the norms of the first $k_0 - 1$ powers reveals that a node is not 1-bounded, the numerical expensive bound has not to be computed.

4.3.3 Upper bounds for norms implemented

By construction, the spectral radius of the matrix corresponding to a strong generator equals 1. When computing the spectral radius and scaling the family, numerical errors occur. But even a little perturbation of the leading eigenvalue can change the situation completely because the generator matrix powers tend to zero or to infinity. For the implementation, it is therefore crucial to substitute the computed spectral radius by the exact value 1.

Falsifying that a node is (strictly) 1-bounded is sufficient to exclude backtracking. In case of the bounding eigenvalue approach, we stop the computation of an upper bound as soon as a vertex with value ≥ 1 is found. When using vectors of balls, we perform a pre-check before computing the extent of the whole vector: The upper bound cannot be smaller than 1 if the extent of the first ball or the extent of the last ball has a value greater or equal to 1.

When computing the upper bound as proposed in Section 4.3.1, we have to balance the length of the ball vector $\mathbf{A}_{J,k_0}$ and the tightness of the upper bound. This is guided by the user-defined parameters LimRadComput, LimRadConstr and MaxK0 as follows.
A ball vector $\mathbf{A}_{J,k_0}$ is constructed such that $r_J(k_0) \leq$ LimRadConstr: To this purpose, we determine k_0 by

$$k_0 \leftarrow \left\lceil \log_a \left(\frac{\text{LimRadConstr}}{\sum_{i=\ell+1}^{d} \|T_i\|} \right) \right\rceil$$

with a being the absolute value of the subdominant eigenvalue of A_J. Choosing LimRadConstr very small simplifies finding a ball $(C, \text{LimRadComput})$ that contains $\left(A_J^\infty, r_J(k_0)\right)$. To prohibit extremely long ball vectors during this step, the user bounds the length with the parameter Maxk0. That is, if $k_0 >$ Maxk0 was determined, then k_0 is set to Maxk0 such that the computed vector $\mathbf{A}_{J,k_0}$ has a length of at most Maxk0. As illustrated in Figure 4.6 (*left*), a ball $\left(A_J^{\tilde{k}}, \text{LimRadComput}\right)$, $\tilde{k} < k_0$, containing $\left(A_J^\infty, r_J(k_0)\right)$ is computed. When doing so, $\tilde{k}$ is chosen such that the number of contained powers A^k, $k < k_0$, is maximized. If there is no such ball, the further computation is done with the vector $\mathbf{A}_{J,k_0}$. Otherwise, a vector of the non-contained balls $(A^k, 0)$, $k < k_0$ and $\left(A_J^{\tilde{k}}, \text{LimRadComput}\right)$ is built.

It may seem at first sight somehow paradox to first compute a — potentially very long — vector for the small radius LimRadConstr and then to quit that accuracy when doing further computation steps with radius LimRadComput. A comparison of Figure 4.6 (*left*) and (*right*) shows the benefit. This 2-step-process allows us to shift the midpoint and therewith reduce the length of the ball vector that would have occurred if we had directly build up a vector with the desired radius LimRadComput.

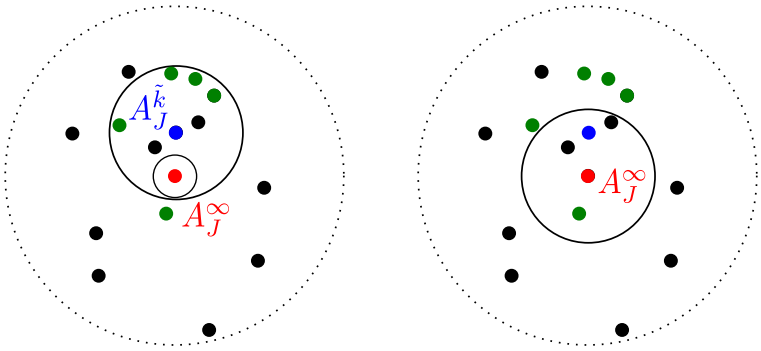


Figure 4.6: To illustrate the benefit of the 2-step-process, $\left(A_J^{\tilde{k}}, \text{LimRadComput}\right)$ containing $\left(A_J^\infty, r_J(k_0)\right)$ (*left*) is opposed to $\left(A_J^\infty, \text{LimRadComput}\right)$ (*right*). Powers A_J^k that are contained in the non-trivial ball in one but not both cases are highlighted in green. In case of this illustrating example, the resulting ball vector after the 2-step-process has length 9 (*left*) in contrast to length 12 (*right*).

4.4 Choosing a norm

It is obvious that the norm has an important impact on the shape of T_* since 1-boundedness is a backtrack criterion of Algorithm 4.3. If $\hat{\rho}(A) = 1$ holds, an extremal norm leads to a J-complete tree with depth 1. But the computation of such a norm is equally difficult, see Chapter 5. Since the set-valued tree approach allows any submultiplicative norm, there is some potential to optimize the runtime

of the algorithm by an appropriate choice of the norm. The examples in Chapter 7 are computed with different norms, sometimes with several to compare the resulting $\mathscr{J}$-complete trees. Many of these examples can be handled with norms that are implemented in MATLAB, as $\|\cdot\|_1$, $\|\cdot\|_2$, or $\|\cdot\|_\infty$. It depends on $\mathscr{A}$ which one is the best to choose.

Norms that are adapted to the matrix family can lead to very trim and short trees such that the computational effort for finding the tree is reduced, at the expense of finding the norm. A very simple idea is to choose, if possible, a weighted norm $\|A\|_S = \|S^{-1}AS\|$ such that $S^{-1}A_JS$ is diagonal for a strong generator J and $\|\cdot\|$ is an axis-oriented[6] norm. If A_J is diagonalizable and S contains its eigenvectors, $\|A_J\|_S = 1$. Obviously, this does not imply that nodes in other branches have small norm values, especially if J is not dominant. In particular, the symmetry of the problem in case of palindromic matrices is destroyed such that Proposition 6.10 is not applicable anymore, see also Chapter 6. The current implementation allows to set a flag indicating if the basis transform by S shall be done.

A closely related but refined approach is pursued by the definition of the norm $\|\cdot\|_V$ in Section 3.3. If we conjecture that $\mathscr{A}$ has a spectral gap at 1, it seems reasonable to base the search on $\|\cdot\|_V$: In case that our conjecture is true and $\mathscr{A}$ is product bounded[7], Theorem 3.15 guarantees a $\mathscr{J}$-complete tree with all nodes being completely positive. In other words, Algorithm 4.2 with initial search tree $\tilde{T} = \{I \in \mathscr{I}_k : k \leq \text{MAXLEVEL}\}$ terminates without updating $\tilde{T}$ when choosing MAXLEVEL sufficiently large.
As it is defined, $\|\cdot\|_V$ depends on a product bounding constant. The computation of this constant is possible, see [Pro96], but experience shows that good results can be achieved if we interpret the constant as a freely selectable parameter w. Slight modifications of the corresponding norm lead to an easy-to-implement variant $\|\cdot\|_w$. Instead of searching for a tree with only completely positive nodes, we target a $\mathscr{J}$-complete tree with exactly one negative node. Therewith, a test on equality to 1 in the nodes J^k is avoided. Conjecturing that $\mathscr{A}$ has a spectral gap at 1, such a norm has the advantage that the number of search trees to consider is dramatically lowered. The decision routine gets the input NEGLIMIT $= 1$ and specifies a child to be negative only in the path $([J^k])_k$. Furthermore, the computation of a tree with mainly completely positive nodes is very efficient. The downside is that, although finiteness is guaranteed for sufficiently large w, other branches might get very long. Hence, a high value of MAXLEVEL may be required to find the 1-bounded leaf and eventually a $\mathscr{J}$-complete tree, which often is not very trim, see also the examples in sections 7.2, 7.3 and 7.5. It might seem intuitive to choose w large as it replaces the product bounding constant. Figure 7.16 and Figure 7.12 demonstrate that this is not always recommendable.

To avoid the shortening of some branches to the expense of others, the adapted norm should take all matrices of the family $\mathscr{A}$ into account. Generating the unit

[6] We call $\|\cdot\|$ axis-oriented if $\|\operatorname{diag}(\lambda_1,\ldots,\lambda_d)\| = \max_{1\leq i\leq d}|\lambda_i|$.
[7] If $\mathscr{A}$ has a spectral gap at 1 and $\mathscr{A}$ is *not* product bounded, $\mathscr{A}$ is reducible.

ball of a norm based on all products up to a certain length seems a natural idea. For a family of real matrices, the computation of a polytope norm $\|\cdot\|_{\text{poly}}$ is described in the following. The process resembles the computation of extremal polytope norms, see Chapter 5. Since we do not target an extremal norm, we end the process after an a priori specified number of iterations IT. Heuristically, this is a good compromise between optimizing the norm and searching the tree. The following algorithm defines the unit ball for $\|\cdot\|_{\text{poly}}$ adapted to $\mathscr{A}$.

The convex hull of a finite set $\mathscr{V} \subset \mathbb{R}^d$ is a bounded and convex polytope being the intersection of finitely many half-spaces. So, there exists $C \in \mathbb{R}^{\ell \times d}$ and $b \in \mathbb{R}^\ell$ with $\ell \in \mathbb{N}$ minimal such that $\text{conv}(\mathscr{V}) = \{x \in \mathbb{R}^d : Cx \leq b\}$. We use this notation in the following algorithm.

Algorithm 4.11 Unit ball construction

Input: real family $\mathscr{A} = \{A_1, \ldots, A_m\}$
set of start vectors V_0
maximal number of iterations IT

Output: set of vertices $\mathscr{V} \subset \mathbb{R}^d$
matrix N such that $\text{conv}(\mathscr{V}) = \{x \in \mathbb{R}^d : Nx \leq 1\}$

Start with $k := 1$

1) $X_k \leftarrow \{A_i x : i \in \{1, \ldots, m\}, x \in V_{k-1}\}$

2) $V_k \leftarrow \{v : v \text{ vertex of } \text{conv}(X_k \cup V_{k-1})\}$

3) $k \leftarrow k + 1$

4) if $k \leq$ IT go to 1)

5) $\mathscr{V} \leftarrow V_k \cup -V_k$

6) determine C, b as defined above

7) if 0 is an entry of b, re-start with modified V_0, otherwise go to 8)

8) $N \leftarrow \text{diag(b)}^{-1} \cdot C$

9) return $\mathscr{V}$ and N

Remarks:
There are sophisticated ways to choose V_0. The extremal norm approaches referenced to in Chapter 5 treat this topic in detail. Since we do not target an extremal norm, this is neglected in the current implementation. The computation of step 2) and 6) make use of the build-in MATLAB function convhulln. C and b are computed in step 6) with use of vert2con.m from MATLAB file exchange[8].
Let $\mathcal{N} := \{n \in \mathbb{R}^d : n^T \text{ is row of } N\}$. We define for $x \in \mathbb{R}^d$

$$\|x\|_{\text{poly}} := \max_{n \in \mathcal{N}} n^T x = \max_i (Nx)_i. \tag{4.10}$$

This is a norm on $\mathbb{R}^d$: Step 5) induces point symmetry such that $v \in \mathcal{V} \iff -v \in \mathcal{V}$. Since $n^T \cdot v = 1 \iff (-n)^T \cdot (-v) = 1$, this implies $n \in \mathcal{N} \iff -n \in \mathcal{N}$. Therewith, positivity of (4.10) is guaranteed. Furthermore, $\|x\|_{\text{poly}} = 0$ only if $x = 0$ since $\{x : Nx \leq 1\}$ is bounded by construction. Absolute homogeneity and triangle inequality are easily verified.

Define V as a matrix such that v is a column of V if and only if $v \in \mathcal{V}$. For the induced matrix norm $\|\cdot\|_{\text{poly}}$ holds

$$\|A\|_{\text{poly}} = \max_{i,j} (NAV)_{ij}. \tag{4.11}$$

To see this, note that $\max_{\|x\|_{\text{poly}}=1} \|Ax\|_{\text{poly}}$ is obtained in an extremal point of $\text{conv}(\mathcal{V})$ since $\|\cdot\|_{\text{poly}}$ is convex. Therewith,

$$\|A\|_{\text{poly}} = \max_{v \in \mathcal{V}} \|Av\|_{\text{poly}} = \max_{v \in \mathcal{V}} \max_i (NAv)_i = \max_{i,j} (NAV)_{i,j}.$$

4.5 A variant of the algorithm to establish contractivity

A variant of Algorithm 4.2 can prove the existence of a contractive tree: Due to Lemma 3.5, a depth-first search on the infinite tree $T = \mathcal{I}$ with backtracking in a strictly 1-bounded node terminates for $\mathcal{A} = \{A_1, \ldots, A_m\}$ if and only if $\hat{\rho}(\mathcal{A}) < 1$. Introducing a maximal search level MAXLEVEL enforces termination in case that $\hat{\rho}(\mathcal{A}) \geq 1$. If the algorithm terminates due to reaching the maximal search level, T_* is not contractive and the output is "no result". We cannot deduce that the family is not contractive since there might be a contractive tree whose depth exceeds MAXLEVEL.
For Algorithm 4.12, issues as a decision routine or the computation of upper bounds for norms are irrelevant. But the chosen norm determines the shape of a contractive tree and therefore influences the output.

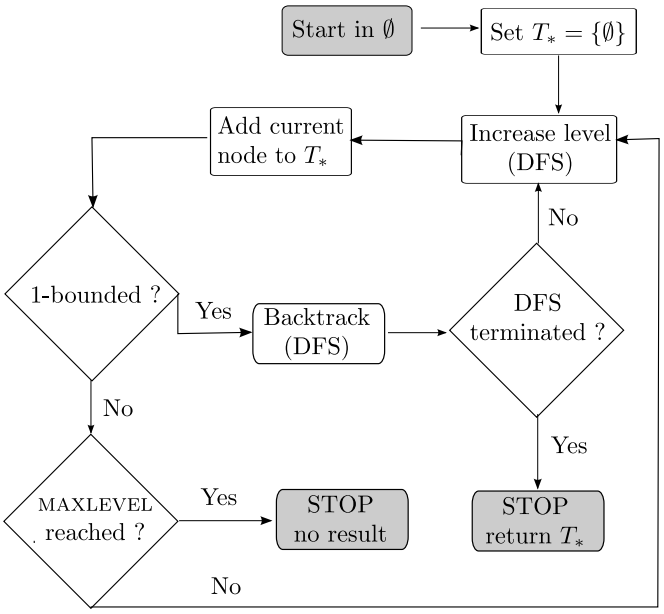
[8] submitted by Michael Kleder, Delta Epsilon Technologies, LLC, 22 Jun 2005 (Updated 11 Jul 2005)

Algorithm 4.12 Search for contractive tree

Input: $\mathscr{A} = \{A_1, \ldots, A_m\}$
submultiplicative matrix norm
maximal search level MAXLEVEL

Output: contractive tree T_* or "no result"

Traverse $\tilde{T} = \{I \in \mathscr{I} : |I| \leq \text{MAXLEVEL}\}$:



4.6 Saving and visualization

The program package includes functions to save and visualize the output tree. To this aim, a MATLAB class NODE was created. The class properties allow to assign to an object NODE the type (1-bounded, covered, negative, or else), its norm, its children, the number of negative entries and the number and types of generator edges that occur in its subtree. The implementation makes use of this class such that saving the root suffices to keep all information that are necessary to reconstruct the tree and collect information about particular nodes a posteriori. There are class functions implemented that allow for a certain node, and in particular for the root, to plot its subtree, to compute the depth of its subtree and to count the number of nodes of its subtree. It is possible to query if a certain index vector is a node of the output tree, and if it is, to access the corresponding NODE object.

5 Related work

The computation of the JSR became a topic of intense research in recent years. Section 5.1 presents some results from the literature concerning computability, which show that the JSR is at least theoretically extremely hard to determine or even to approximate. Nevertheless, there exist a few algorithms that are useful in practice. Roughly speaking, there are two lines of research. One of them attempts to compute a norm adapted to the matrix family in order to achieve close or even coinciding bounds in the three-member-inequality. The other one more or less neglects the choice of norm and bases on the convergence of bounds for increasing product length k, pursuing graph-theoretical ideas on the tree of matrix products. Both approaches lead to algorithms of approximation, the first one in addition proved to be useful for establishing the FP and therewith determining the exact value of the JSR. The set-valued tree method can be put into the context of the graph-theoretical one, but also aims to validate the FP.
Without claim to completeness, we give a short overview on existing methods for approximation in Section 5.2 and for exact determination in Section 5.3. The contractivity of a matrix family, though not the main focus of this work, is a problem of some importance in applications and shortly taken up in Section 5.4.

5.1 Computability

In some cases, the JSR of a matrix family $\mathscr{A}$ is easily determined: If, by a similarity transformation, $\mathscr{A}$ can be transformed simultaneously to normal, symmetric, hermitian or triangular matrices, then the JSR equals the largest spectral radius of the matrices in $\mathscr{A}$. Proofs or references can be found in [Jun09].
For certain classes of pairs of (2×2)-matrices, there exist explicit formulas for the joint spectral radius, see [Möß10, BZ00, LR94, GMW94]. [Gri96] provides a condition for a finite family $\mathscr{A}$ which is sufficient to conclude that $\hat{\rho}(\mathscr{A}) = \max\{\rho(A) : A \in \mathscr{A}\}$.

For a family $\mathscr{A}$ of nonnegative matrices without entries between zero and one, there exists a polynomial time algorithm to decide whether $\hat{\rho}(\mathscr{A}) < 1$, $\hat{\rho}(\mathscr{A}) = 1$ or $\hat{\rho}(\mathscr{A}) > 1$, see [Jun09]. Without this restriction on the matrix entries, the situation is different: If the family consists of matrices with real or even with rational entries, determining if $\hat{\rho}(\mathscr{A}) \leq 1$ is, according to [BT00], a Turing-undecidable problem. Hence, no algorithm exists in general that determines the joint spectral radius in finite time.
The results for an approximation are discouraging as well. We say that a value ρ^* approximates the JSR $\hat{\rho}$ with *relative accuracy* $1 - \epsilon$ if

$$(1 - \epsilon)\rho^* \leq \hat{\rho} \leq \rho^*$$

and the *relative error* is ϵ. According to [BN05], there exists, unless P=NP, no approximation algorithm which is polynomial in both the dimension of the matrices d and $1/\epsilon$.

5.2 Approaches to approximation

In principle, the three-member-inequality displayed in Section 2.2 allows to approximate the JSR with arbitrary accuracy. Let T be the m-ary set-valued tree T that results from $\mathscr{J} = \emptyset$, i.e., the nodes in level k code the products A_I with $I \in \mathscr{I}_k$. Performing a breadth-first search on T, every level provides lower and upper bounds which converge towards the JSR. The costs are exponentially increasing with the level, since m^k products are involved in level k.
Gripenberg proposes in [Gri96] a branch and bound algorithm which is a modification of this method. The idea is to discard branches that start in nodes whose norms are too small in some sense compared to the spectral radii of the already visited nodes. The products coded by these nodes are known not to determine the supremum in Definition 2.1. Fix $\delta > 0$ and consider in level k the recursively defined set of nodes

$$S_k = \left\{[I,i] : I \in S_{k-1}, i \in \mathscr{I}_1, \|A_I\| \geq \alpha_{k-1} + \delta\right\}$$

with $\alpha_k = \max\left\{\alpha_{k-1}, \sup_{I\in S_k} \rho(A_I)^{\frac{1}{k}}\right\}$, $S_1 = \mathscr{I}_1$ and $\alpha_1 = \max_{I\in\mathscr{I}_1} \rho(A_I)$. With $\beta_1 = \max_{I\in\mathscr{I}_1} \|A_I\|$ and

$$\beta_k = \min\left\{\beta_{k-1}, \max\left\{\alpha_k + \delta, \sup_{I\in S_k} \rho(A_I)^{\frac{1}{k}}\right\}\right\},$$

upper and lower bounds for the JSR are given by $\alpha_k \leq \hat{\rho}(\mathscr{A}) \leq \beta_k$ for each $k \in \mathbb{N}$. Furthermore, $\lim_{k\to\infty}(\beta_k - \alpha_k) \leq \delta$. Adapting δ, the method is capable to find arbitrary small enclosing intervals.

An independent idea to improve the efficiency of the breadth-first search on T, which can be combined with Gripenberg's approach, is presented by Maesumi in [Mae96]. Instead of calculating the spectral radii of m^k products of length k, the same result is obtained by computing no more than m^k/k matrices. Maesumi shows that the spectral radius is invariant under cyclic permutation of the index vector, i.e., $\rho(A_I) = \rho(A_{\pi(I)})$. Furthermore, if $\rho(A_I)$ is determined, spectral radii of products $A_{I^\ell}, \ell \geq 1$ are given without further matrix computations by $\rho(A_{I^\ell}) = \rho(A_I)^\ell$. Elimination of $\Pi(I)$ leads to $\frac{m^k}{k}$ remaining nodes in level k.

There is an obvious problem when applying this type of algorithm. To satisfy $\beta_k - \alpha_k \leq \epsilon$ for some $\epsilon > 0$, k might have to be chosen very large. Due to [GP13], the number k grows as C/ϵ, with ϵ being the relative error of the approximation and $C > 0$ a constant that might be large for high dimensions d. Additionally,

the number of products might be exponentially increasing despite of the proposed modifications.

In contrast, [Pro05] presents an algorithm for approximation with relative error ϵ which is, for fixed dimension d, polynomial with respect to $\frac{1}{\epsilon}$. This geometric approach accounts for irreducible families of real matrices. It is shown that such a family always possesses an *invariant body* $M \subset \mathbb{R}^d$, that is, M is convex, compact, with non-empty interior, centrally symmetric with respect to the origin such that

$$\operatorname{conv}(A_1 M, \ldots, A_m M) = \lambda M$$

for some $\lambda \in \mathbb{R}_+$. The invariant body of $\mathscr{A}$ is not unique. But for any invariant body, we have

$$\lambda = \hat{\rho}(\mathscr{A}).$$

Constructing iteratively polytopes, an invariant body is approximated. The algorithm terminates after $N = c(\mathscr{A}) \cdot \epsilon^{-1}$ steps, with $c(\mathscr{A})$ being a constant that can be computed from the eigenspaces of $\mathscr{A}$. The desired approximation of $\hat{\rho}(\mathscr{A})$ is given by $n^{\frac{1}{N+1}}$ with $n := \max_{v \in \mathscr{V}} \|v\|_2$ and $\mathscr{V}$ the vertices of the polytope constructed in the last iteration.

For fixed relative accuracy $1 - \epsilon$, [BN05] provides an algorithm that runs in time polynomial in $d^{\ln(m)/\epsilon}$, with m being the number of matrices. The approximation bases on the following idea: If the matrix family $\mathscr{A}$ possesses a proper invariant cone[1], then lower and upper bounds for $\hat{\rho}(\mathscr{A})$ are given by spectral radii of certain sums of Kronecker products of $\mathscr{A}$. Introducing a so-called semidefinite lifting, a proper invariant cone can be enforced such that a combination of semidefinite lifting and generation of Kronecker products leads to the desired approximation for the JSR of the original family.

To improve the computational complexity for high dimensions d, [PJB10] proposes a method in the same spirit that avoids the generation of Kronecker products. By a semidefinite lifting, a matrix family with common invariant cone K is attained. Then upper and lower bounds for the joint spectral radius are given for each $k \in \mathbb{N}$ by the so-called joint conic radius of $\{A_I : I \in \mathscr{I}_k\}$, which depends on K and is efficiently computable in the framework of conic programming. An approximation with relative error ϵ is obtained for $k \geq (\ln 1/\alpha) \cdot \epsilon^{-1}$, with α being the largest number such that for any compact convex set $G \subset K$ there exists $v \in G$ for which $\frac{1}{\alpha} v - G \in K$.

There exist further approaches to approximate the JSR, as by computation of an ellipsoidal norm [AS98, BNT05] or by using polynomial sums of squares [PJ08].

[1] For the definition of a proper invariant cone, see [BN05].

5.3 Approaches to exact determination

Rota and Strang showed in [RS60] that the JSR of a bounded family can equivalently be defined by

$$\hat{\rho}(\mathscr{A}) = \inf_{\|.\|} \max_{A\in\mathscr{A}} \|A\|. \tag{5.1}$$

If the infimum is attained for a norm $\|\cdot\|_*$, it is called an *extremal norm*[2]. A family $\mathscr{A}$ of complex matrices possesses an extremal norm if and only if $\mathscr{A}$ is *non-defective*, which means that the family $\hat{\rho}(\mathscr{A})^{-1}\cdot\mathscr{A}$ is product bounded. Hence, by Elsner's lemma[3], irreducible families always have an extremal norm. If A_J is an FP-product, then the coinciding bounds of the three-member-inequality

$$\rho(A_J)^{\frac{1}{|J|}} = \max_{A\in\mathscr{A}} \|A\|_*$$

determine the JSR. Therewith, $\rho(A_J) = 1$ implies that a norm with $\max_{A\in\mathscr{A}}\|A\| = 1$ is extremal and that A_J is FP-product. In terms of the set-valued tree approach, the trivial tree with nodes $\{\emptyset\}\cup\mathscr{I}_1$ is $\mathscr{J}$-complete for $\mathscr{J} = \{J\}$ with respect to this norm.

This motivates the methods presented in the following. Analogous to the set-valued tree method, one first conjectures an FP-product A_J and scales the family $\mathscr{A}$ as described in Section 2.4 such that J is a generator of the scaled family. In contrast to our approach, the common idea is to determine an extremal norm by computing the corresponding unit ball based on A_J. If the computation stops after finite time, a unit ball is established, proving that the generator matrix A_J is an FP-product.

The approach in [Mae00, Mae08] is targeted on constructing a special extremal norm, the so-called *optimal* norm. For any non-trivial bounded set S, such a norm is induced by the unit ball being the intersection of all extremal unit balls containing S. The iterative computation of such a unit ball starts with determining the invariant ball of A_J being the maximal set G with $A_J G = G$ and $\sup_{x\in G}\|x\| < 1$ for some vector norm $\|\cdot\|$. Set $G_q = \operatorname{conv}\left(G_{q-1}\cup\mathscr{A}G_{q-1}\right)$ with $G_0 := G$. If $G_q = G_{q-1}$ for some q, then G_q is the optimal unit ball and A_J is proven to be an FP-product. [Mae08] introduces a negative exit for the algorithm in case that A_J is no FP-product, giving a criterion involving certain interior points of G_q.

[GWZ05] introduces the so-called complex polytope norms to obtain an extremal norm. Let $\operatorname{absco}(X)$ be the set of all finite absolutely convex combinations of vectors in $X\subset C^d$, i.e.,

$$\operatorname{absco}(X) := \left\{ x\in\mathbb{C}^d : x = \sum_{i=1}^{\ell}\lambda_i x_i \text{ with } \lambda_i\in\mathbb{C}, x_i\in X \text{ and } \sum_{i=1}^{\ell}|\lambda_i|\le 1 \right\}.$$

[2] Even singleton families $\mathscr{A} = \{A\}$ do not necessarily possess an extremal norm.
[3] see Section 2.3

A bounded set $P \subset \mathbb{C}$ is called a *balanced complex polytope* (b.c.p.) if there exists a finite set $\mathcal{V} = \{v_1, \ldots, v_\ell\}$ spanning $\mathbb{C}^d$ such that $P = \text{absco}(\mathcal{V})$. Any b.c.p. is the unit ball of a so-called *complex polytope norm*. It is shown that the set of norms to be considered in (5.1) can be limited to the set of all possible induced complex polytope norms and conjectured that any non-defective family that has the FP possesses an extremal complex polytope norm. The authors were able to proof a result with a quite restrictive condition on the matrix family. They call a family $\mathscr{A}$ *asymptotically simple* if it has a minimal[4] FP-product A_J with only one leading eigenvector such that the set

$$\{x : x \text{ is leading eigenvector of } A_{J'}, J' \in \pi(J)\}$$

equals the set of leading eigenvectors of all finite or infinite[5] FP-products. A sufficient condition for the existence of an extremal complex polytope norm is given by the *small complex polytope extremality (CPE) Theorem*: Consider an asymptotically simple family with leading eigenvector x and denote by $T(\mathscr{A}, x)$ the trajectory of x under the scaled family $\mathscr{A}$. If $T(\mathscr{A}, x)$ spans $\mathbb{C}^d$, then $S(\mathscr{A}, x) := \text{absco}(T(\mathscr{A}, x))$ is a b.c.p. Furthermore, due to [GZ08], if $T(\mathscr{A}, x)$ is additionally a bounded subset of $\mathbb{C}^d$, then $\mathscr{A}$ is non-defective, $S(\mathscr{A}, x)$ is the unit ball of an extremal complex polytope norm and $\hat{\rho}(\mathscr{A}) = 1$. [GZ08] presents an algorithm for the construction of $S(\mathscr{A}, x)$, with x being the leading eigenvector of a generator matrix. In case that the algorithm terminates after a finite number of steps, $S(\mathscr{A}, x)$ is a b.c.p. and if the vertices span $\mathbb{C}^d$, then the according complex polytope norm is extremal. That is, $\hat{\rho}(\mathscr{A}) = 1$ is verified.
Obviously, asymptotic simplicity leads to strong restrictions for the eigenvalues of the FP-product. In particular, it implies that an FP-product of a real matrix family is not allowed to have a complex conjugate pair of leading eigenvalues. To extend the small CPE theorem to this case, the definition of asymptotic simplicity is relaxed in [GZ09] such that the minimal FP-product is allowed either to have a unique real leading eigenvector or to possess a unique pair of complex conjugate leading eigenvectors. Furthermore, the algorithm from [GZ08] is modified for real families to incorporate the new result.

Even the relaxed definition of asymptotic simplicity is a strong limitation for the matrix family but it is a necessary assumption for the existence of a b.c.p. constructed as described above. [GZ08] exhibits two examples which are not asymptotically simple and for which no b.c.p. can be found by the algorithm [GZ08]. The set-valued tree method, in contrast, settles both problems, see Chapter 7. The corresponding $\mathscr{J}$-complete trees are even rather slim.

By computing extremal real polytope norms for all pairs of 2×2 sign-matrices, [CGSCZ10] established the finiteness property for this class of matrices. This is an interesting result since, due to [Jun09], the finiteness property holds for all finite

[4] Minimal means here that A_J is not a power of another FP-product.

[5] To be precise, we defined FP-products for finite length. See [GWZ05] for the generalized notion for infinite products, called l.s.m.p.

sets of rational matrices if and only if it holds for all pairs of sign-matrices with arbitrary dimension.

[GP13] modifies the complex polytope approach developed in [GWZ05, GZ08, GZ09] to be able to deal with higher dimensions d. Instead of generating the unit ball from the trajectory of the leading eigenvector x_J of one FP-product A_J, all leading eigenvectors $x_{J'}$ of cyclic permutations $A_{J'}$, $J' \in \pi(J)$ are taken into account. The authors define a graph, called *cyclic tree*, whose nodes are vectors from the trajectories $T(\mathscr{A}, x_{J'}), J \in \Pi(J)$ and whose edges symbolize multiplication with elements of $\mathscr{A}$ from the left. They avoid the computation of arbitrary powers of cyclic permutations of the generator in an elegant way by defining the root of the cyclic tree to be a cycle of the vectors $x_{J'}, J' \in \Pi(J)$. The paths in the tree starting in the root cycle reduce to suffixes whose corresponding prefixes are elements of $\Pi(J)$. In other words, all trajectory points $A_{[J'^k,S]}x_{J'}$, $k \in \mathbb{N}_0$ are described by the node $v = A_S x_{J'}$. The complex polytope of the k-th iteration can be computed by knowledge of the nodes up to level k of the tree. The computation is made more efficient by saying that a node coding a vector from the interior of the current polytope is a *dead leaf*, and the branch can be cut. If there exists a level with only dead leaves, then the b.c.p. does not change with respect to the iteration before. That is, this b.c.p. is the unit ball of an extremal norm and the algorithm terminates. If the algorithm does not stop after a certain number of steps, the complex polytope norm computed in the last iteration is used for approximating the JSR by the three-member-inequality. Furthermore, a stopping criterion is introduced that reveals if a generator matrix is no FP-product. See [GP13] for a detailed description of the cyclic trees and the resulting algorithm. Variants allowing a more efficient computation in case of real and nonnegative matrix families base on the same concept.

Clearly, the cyclic tree differs from the set-valued tree in many aspects. Nevertheless, both depend in their combinatorial structure of the generator J. The cyclic root of the tree has the same functioning as the introduction of negative nodes, namely the subsumption of an infinite set of generator powers in order to obtain a finite structure. The set-valued tree approach subsumes only the powers of J and in general not additionally those of $\pi(J)$ since the powers J^{k+1} and $\pi(J)^k$ differ only in terms of a finite prefix and suffix.
Although the algorithm of [GP13] is theoretically applicable to all complex matrix families, it is capable to determine the exact value of the JSR if and only if the family has a spectral gap at 1: The algorithm terminates after finitely many iterations if and only if J is a dominant generator for $\mathscr{A}$. This is an even stronger restriction than asymptotic simplicity, which principally allows several strong generators. For the set-valued tree approach, Theorem 3.15 states that a $\mathscr{J}$-complete tree exists always if $\mathscr{A}$ has a spectral gap at 1. Choosing an appropriate norm, this case is in fact trivial because the nodes of the tree can all be chosen to be completely positive, i.e., no infinite sets of matrices have to be employed. Several examples in Chapter 7 show that it is possible to establish the JSR of families which do *not* have a spectral gap at 1. Despite possible principal advantages, we have to acknowledge

that typical run-times of our current implementation can hardly compete with the impressive results presented in [GP13].

[JCG14] combines the semidefinite lifting method of [BN05, PJB10] with the iterative construction of a polytope norm [GWZ05, GZ08, GZ09] by searching for an *extremal conitope*, which provides an extremal norm for the lifted family. This validates the FP if the algorithm terminates in finite time. Two different but closely related algorithms are presented. One of them attempts to verify that a generator is FP-product, stopping after a finite number of steps either confirming the FP or with an approximation of the JSR. The other one is an approximation algorithm that aims to achieve rapidly converging bounds. The sufficient condition for existence of an extremal conitope norm is inherited from the one of an extremal polytope norm. If a family is irreducible and asymptotically simple, then the lifted matrix family admits an extremal conitope norm.

In the context of smoothness analysis of subdivision schemes, [Rio92] discusses three different methods to determine upper[6] bounds for the JSR of a pair of matrices. The first one corresponds to the breadth-first search with respect to $\|\cdot\|_\infty$ on the tree of products and therewith providing in general an approximation of the JSR by the three-member-inequality. The second one refers to [DL92b] and is an attempt to construct an extremal norm of $\mathscr{A} = \{A_1, A_2\}$ but can by design be successful only if $\hat{\rho}(\mathscr{A}) = \max_{i=1,2} \rho(A_i)$. Assume that $\rho(A_1) \geq \rho(A_2)$ and B is a matrix whose columns are eigenvectors of A_1. The idea is to find a parametrization of B by d numbers, one for each column, and a matrix norm such that $\|B^{-1}A_2B\| \leq \rho(A_1)$. In that case, $\hat{\rho}(\mathscr{A}) = \rho(A_1)$ holds due to $\|B^{-1}A_1B\| = \rho(A_1)$. The third method accounts for a special class of matrix families, that is, subdivision matrices with a strictly linear phase mask, i.e., the mask is symmetric and the Fourier transform of the scheme's symbol $b\left(e^{-i\xi}\right)$ is positive for $\xi \in [-\pi, \pi]$. An upper bound for the JSR can be computed by determining the spectral radius of some *single* matrix, avoiding a joint spectral radius analysis. Under some additional conditions, the upper bound is sharp. This approach is reviewed and refined in [FM12]. Weaker conditions for optimality of the bound are shown and an alternative matrix of a smaller dimension is constructed whose spectral radius is to be computed.

5.4 Contractivity

Many applications rather need to check whether $\hat{\rho}(\mathscr{A}) < 1$ than to obtain the exact or approximated value. If $\mathscr{A}$ has the finiteness property, this problem is decidable [Jun09]: Perform a breadth-first search on the tree of products T and compute for each level the bounds of the three-member-inequality. If there is a level such that the lower bound equals or exceeds 1, stop and declare $\hat{\rho}(\mathscr{A}) \geq 1$ and if the upper bound falls below 1, stop and declare $\hat{\rho}(\mathscr{A}) < 1$. This algorithm

[6] Upper bounds of the JSR transform to lower bounds for Hölder regularity of a scheme, see Chapter 6.

terminates since the case $\hat{\rho}(\mathscr{A}) = 1$ and lower bounds being strictly smaller than 1 for all k is excluded by the finiteness property.

To validate $\hat{\rho}(\mathscr{A}) < 1$, it is actually not necessary to perform a breadth-first search. The number of products to consider is dramatically reduced by traversing the tree with a depth-first search, see also [HMR09] and references therein.

[HMR09] considers a parametrized family $\mathscr{A}_\omega$ that results from the smoothness analysis of a subdivision scheme, the generalized 4-point scheme proposed in [DLG87], and determines explicitly the interval of parameters with $\hat{\rho}(\mathscr{A}_\omega) < 1$. Although the approach was adapted to this concrete matrix family, the set-valued tree method bases strongly on the underlying ideas. The tree defined in [HMR09] corresponds to the set-valued tree with $\mathscr{J} = \emptyset$, which is traversed by depth-first search with backtracking in a strictly 1-bounded node. The notion of a generator[7] is established to deal with long periodic paths. The observation that generators induce for the scaled family an infinite periodic path in the tree of all products inspired the idea of subsuming such a path by introducing negative children. With some modifications, some of the proof techniques turn out to be very useful for our purpose as well, as for example the idea of calculating the limit of generator matrix powers A_J^∞ or the splitting of $I \in \mathscr{I}$ into nodes of the tree, which motivated the I_*-maximal prefix partition of $K \in \mathscr{K}$.

[7] Note that a generator set in [HMR09] is defined differently than in this work, the main idea though is the same.

6 Joint spectral radius in subdivision

The analysis of subdivision schemes is one of many mathematical topics involving the JSR. There are many different classes of subdivision schemes. To exemplarily demonstrate an application of the set-valued tree method, we concentrate on the most simple class of schemes, that is, on univariate, linear, stationary, uniform, compactly supported subdivision schemes. Section 6.1 aims to shortly introduce the general idea of subdivision schemes, to clarify the notation and to recall some key results of smoothness analysis. Section 6.2 explains in detail how matrices can be derived from a scheme in order to analyze its regularity. In particular, an explicit formula for schemes of arbitrary arity m is provided. In Section 6.3, the specific properties of palindromic subdivision matrices, which occur in case of binary symmetric subdivision, are examined with respect to the set-valued tree approach. It is shown that, due to the symmetry of the matrices, considering half of the set-valued tree is sufficient when choosing the norm adequately. As another consequence, the matrices in general do not possess a spectral gap at 1 such that Theorem 3.15 is not applicable. An exception are families with generators of length 2. To handle other cases, the so-called palindromic transformation is developed.

6.1 A short introduction to subdivision

A *subdivision scheme* S is a set of rules which determine how to generate from a given sequence P^k of values in $\mathbb{R}^d$, called *control points*, a denser one P^{k+1}. *Subdivision* is the iterated process of applying these rules, which, as functions of the control points, may be linear or non-linear. One application of the rules is called a *refinement step*, and the control points we started with are referred to as *old points* and the arisen ones as *new points*. The scheme is called *uniform* if the same rules are applied all over the sequence of control points, and *stationary* if the rules do not change from one refinement step to the next. If the number of points doubles/triples/quadruples in one subdivision step, the scheme is called *binary/ternary/quaternary*. More generally, if the number of points multiplies by m, we call it an *m-ary scheme*, and m is the *arity*. It is *symmetric*, if the subdivision step commutes with reversion of the order of control points. The scheme has *compact support* if only finitely many old points contribute to the generation of a new one. Considering *univariate* schemes, the control points are associated in step k with abscissae $m^{-k}\mathbb{Z}$. By linear interpolation of the control points, we obtain a sequence of piece-wise linear functions when applying the subdivision scheme iteratively.

In the following, we assume a subdivision scheme to be univariate, linear, stationary, uniform and compactly supported. Given such a scheme S, obviously the question as to the existence of a limit function arises. If the control points are in $\mathbb{R}^d$, the limit function maps from $\mathbb{R}$ to $\mathbb{R}^d$. The smoothness analysis can be done separately for each component, so we assume in the following control points in $\mathbb{R}$ so that we can restrict ourselves to the analysis of real-valued functions.
Denote by $S^kP^0 : \mathbb{R} \to \mathbb{R}$ the function interpolating $\left(m^{-k}u, P_u^k\right)$ and being linear on $m^{-k}[u, u+1]$ for any $u \in \mathbb{Z}$. If $(S^kP^0)_k$ converges compactly to a continuous function $f : \mathbb{R} \to \mathbb{R}$, we call f the *limit function corresponding to* P^0 and write $f =: S^\infty P^0$. The *basic limit function* of S is given by $\varphi := S^\infty \delta_{u,0}$ and plays an important role in Section 6.2.
S is termed *convergent* if, for any bounded initial sequence P^0, there exists a limit function $S^\infty P^0$ which is nontrivial for at least one sequence of initial data. If the iterated refinement steps map any bounded initial data sequence to zero, S is *contractive*.
Convergence of a scheme imposes the question as to the smoothness of the limit. If the limit function is in C^k for any initial data sequence, S is called C^k or C^k-convergent.
Our main focus is on Hölder regularity of the limit function: For $k \in \mathbb{N}_0$ and $\alpha \in (0, 1]$, let

$$C^{k,\alpha} := \{f \in C^k : \exists c \in \mathbb{R}_{>0} \text{ s.t. } \|f^{(k)}(x) - f^{(k)}(y)\| \le c \cdot \|x - y\|^\alpha\}.$$

That is, a function in $C^{k,\alpha}$ has continuous derivatives up to order k, and the k-th derivative is *Hölder continuous with Hölder exponent* α. Define

$$C_*^r := \{f \in C^{k,\alpha} : k + \alpha < r\}, \quad r \in \mathbb{R}.$$

If $S^\infty P^0 \in C_*^r$ for any initial sequence P_0, we say that S is C_*^r and has *Hölder regularity* r. If S is C_*^r but not $C_*^{r+\varepsilon}$ for any $\varepsilon > 0$, we say that r is the *maximal Hölder regularity* and $\alpha_* := r - \lfloor r \rfloor$ is the *critical Hölder exponent* of S. That is, the critical Hölder exponent is a sharp upper bound for the Hölder exponents of S but it is not necessarily Hölder exponent of S itself.

For the class of schemes considered here, the *refinement rule*

$$P_u^{k+1} = \sum_{v \in \mathbb{Z}} a_{u-mv} P_v^k \tag{6.1}$$

defines S, with m being the arity. The sequence of coefficients $(a_u)_{u\in\mathbb{Z}}$ is called the *mask* of S. Since S has compact support, the mask has finitely many non-zero elements. With $\underline{a} := \min\{i : a_i \neq 0\}$ and $\overline{a} = \max\{i : a_i \neq 0\}$, the mask is represented by the finite vector

$$\mathbf{a} := \left[a_{\underline{a}}, \ldots, a_{\overline{a}}\right]$$

and $\operatorname{supp} \mathbf{a} := [\underline{a}, \overline{a}]$ is its support. The *symbol* of S is the Laurent polynomial $a(z) := \sum_{u\in\mathbb{Z}} a_u z^u$, a useful tool for smoothness analysis. We denote the scheme S by $S_{\mathbf{a}}$ if the dependance of a scheme on the mask resp. symbol shall be indicated.

In the following, we recall some well known results that are crucial for the smoothness analysis of a subdivision scheme. If $S_{\mathbf{a}}$ is convergent, then

$$\sum_{u\in m\mathbb{Z}+\ell} a_u = 1, \quad \ell \in \{0,\dots,m-1\}.$$

This implies for the mth roots of unity $\xi_m^k := e^{\frac{i2\pi k}{m}}$ that

$$a(\xi_m^k) = \sum_{\ell=0}^{m-1}\sum_{u\in m\mathbb{Z}+\ell} a_u(\xi_m^k)^u = \sum_{\ell=0}^{m-1}(\xi_m^k)^\ell \sum_{u\in m\mathbb{Z}+\ell} a_u = \begin{cases} m & \text{for } k=0 \\ 0 & \text{for } k=1,\dots,m-1. \end{cases}$$

Hence, $a(1) = m$ and the divisibility of $a(z)$ by $\sum_{i=0}^{m-1} z^i$ are necessary conditions for convergence of $S_{\mathbf{a}}$ and will be assumed, throughout.

A central method is to analyze the *difference scheme*, a subdivision scheme that relates differences of P^k to differences of P^{k+1}. If $S_{\mathbf{a}}$ is a scheme with symbol $a(z) = \left(\sum_{i=0}^{m-1} z^i\right) b(z)$, then $S_{\mathbf{b}}$ is its difference scheme, see [Sab10]. It allows to characterize convergence of $S_{\mathbf{a}}$ as follows:

Theorem 6.1 *$S_{\mathbf{a}}$ is convergent if and only if $S_{\mathbf{b}}$ with $b(z) = \frac{1}{\left(\sum_{i=0}^{m-1} z^i\right)} a(z)$ is contractive.*

Proof. For $m = 2$, this corresponds to [DL02], Theorem 4.8. The arguments of the proof are analogous for arbitrary arity, displayed below in the generalized variant. It is not difficult to see that $S_{\mathbf{b}}$, being the difference scheme, is contractive if $S_{\mathbf{a}}$ is convergent. To prove the converse, we use the z-transform $L(P^k;z) = \sum_u P_u^k z^u$ of a sequence P^k to show that $(S_{\mathbf{a}}^k P^0)_k$ is, for arbitrary P^0, a Cauchy sequence with respect to the sup-norm if $S_{\mathbf{b}}$ is contractive. This implies that $S_{\mathbf{a}}$ is convergent.
As piecewise linear function, $S_{\mathbf{a}}^k P^0$ attains its extreme values at the breakpoints. Therewith,

$$\sup_{x\in\mathbb{R}} \left|S_{\mathbf{a}}^{k+1}P^0(x) - S_{\mathbf{a}}^k P^0(x)\right| = \max_{i\in\{0,\dots,m-1\}} \left\{ \sup_{u\in\mathbb{Z}} \left|P_{mu+i}^{k+1} - g_{mu+i}^{k+1}\right| \right\},$$

where

$$g_{mu+i}^{k+1} := \frac{m-i}{m} P_u^k + \frac{i}{m} P_{u+1}^k. \tag{6.2}$$

In terms of the z-transform, (6.2) is represented by

$$\begin{aligned} L(g^{k+1};z) &= \sum_{i=0}^{m-1}\sum_{u\in\mathbb{Z}} g_{mu+i}^{k+1} z^{mu+i} \\ &= \sum_{i=0}^{m-1} \left(\frac{m-i}{m} z^i + \frac{i}{m} z^{i-m}\right) \cdot \left(\sum_{u\in\mathbb{Z}} P_u^k z^{mu}\right) \\ &= \sum_{i=0}^{m-1} \left(\frac{m-i}{m} z^i + \frac{i}{m} z^{i-m}\right) \cdot L(P^k;z^m) \\ &= \frac{z^{m+1} - 2z + z^{-m+1}}{m(z-1)^2} \cdot L(P^k;z^m). \end{aligned} \tag{6.3}$$

It is not difficult to see that $L(P^{k+1};z) = a(z)L(P^k;z^m)$ and $L(P^k;z) = (1-z)L(\Delta P^k;z)$ where ΔP^k denotes the sequence of differences $(P^k_{u+1} - P^k_u)_{u\in\mathbb{Z}}$. Together with (6.3), we obtain

$$\begin{aligned} L(P^{k+1};z) - L(g^{k+1};z) &= a(z)L(P^k;z^m) - \frac{z^{m+1} - 2z + z^{-m+1}}{m(z-1)^2} L(P^k;z^m) \\ &= \left(\frac{1-z^m}{1-z} b(z) - \frac{1-z^m}{1-z} \frac{\sum_{i=0}^{m-1} z^{i-m+1}}{m} \right) L(P^k;z^m) \\ &= \frac{b(z) - \frac{1}{m}\sum_{i=0}^{m-1} z^{i-m+1}}{1-z} L(\Delta P^k;z^m). \end{aligned}$$

Let $d(z) := b(z) - \frac{1}{m}\sum_{i=0}^{m-1} z^{i-m+1}$. Since $b(z) = \frac{1}{\left(\sum_{i=0}^{m-1} z^i\right)} a(z)$ and $a(1) = m$, it is $d(1) = b(1) - 1 = 0$. Hence, $e(z) := \frac{d(z)}{1-z}$ is a Laurent polynomial and

$$L(P^{k+1};z) - L(g^{k+1};z) = e(z)L(\Delta P^k;z^m).$$

Therewith, $P^{k+1} - g^{k+1} = S_{\mathbf{e}}\Delta P^k = S_{\mathbf{e}}(S^k_{\mathbf{b}}\Delta P^0)$. Following the argumentation in [DL02], this together with $S_{\mathbf{b}}$ being contractive implies that $S^k_{\mathbf{a}}P^0$ is uniformly convergent. □

In [DL02], it is shown that $\frac{1+z}{2}$ is a smoothing factor in the binary case. That is, if $a(z) = \frac{1+z}{2} b(z)$ and $S_{\mathbf{b}}$ is C^k, then $S_{\mathbf{a}}$ is C^{k+1}. The smoothing factor for arbitrary arity m is specified in [Sab10] to be $\frac{\sum_{i=0}^{m-1} z^i}{m}$. Therefore, further smoothness analysis requires to regard the *divided difference scheme* of some scheme $S_{\mathbf{a}}$, given by $S_{\mathbf{b}}$ with $b(z) = \frac{m}{\sum_{i=0}^{m-1} z^i} a(z)$.

Theorem 6.2 *Let* $a(z) = \frac{\left(\sum_{i=0}^{m-1} z^i\right)^{k+1}}{m^k} b(z)$ *and* $S_{\mathbf{b}}$ *contractive. Then* $S_{\mathbf{a}}$ *is* C^k.

Proof. For $m = 2$, this corresponds to [DL02], Theorem 4.11, [DL02], Corollary 4.21 and [DL02], Corollary 4.22. With the generalized smoothing factor, the claim follows from Theorem 6.1 with similar arguments. □

In general, the implication of Theorem 6.2 is no equivalence. This is only the case if the scheme $S_{\mathbf{a}}$ is *stable*: We follow the definition of [Rio92], saying that $S_{\mathbf{a}}$ is stable if

$$\sum_{n\in\mathbb{Z}} \varphi(n) e^{in\omega} \neq 0 \quad \forall \omega \in \mathbb{R}$$

with φ being the basic limit function of $S_{\mathbf{a}}$. This implies that $S_{\mathbf{a}}$ is L_∞-stable as defined in [DL02]: There are constants $0 < C_1 \leq C_2 < \infty$ such that

$$C_1 \sup_{u\in\mathbb{Z}} |P_u| \leq \left\| \sum_{u\in\mathbb{Z}} P_u \varphi(x-u) \right\|_\infty \leq C_2 \sup_{u\in\mathbb{Z}} |P_u|$$

for any sequence $(P_u)_{u\in\mathbb{Z}}$.

Theorem 6.3 *If $S_{\mathbf{a}}$ is C^k and stable, then $a(z)$ is divisible by $\left(\sum_{i=0}^{m-1} z^i\right)^{k+1}$, and $S_{\mathbf{b}}$ with $b(z) = \frac{m^k}{\left(\sum_{i=0}^{m-1} z^i\right)^{k+1}} a(z)$ is contractive.*

Proof. For $m = 2$, this corresponds to [Rio92], Theorem 9.2 or to a combination of [DL02], Theorem 4.18 and [DL02], Corollary 4.22. Generalizing the smoothing factor, the claim follows with similar arguments basing on Theorem 6.1. □

That is, we have to check the difference scheme of the k-th divided difference scheme for contractivity in order to prove that the original scheme is C^k. Furthermore, Hölder regularity can be derived from that very scheme, as we explain in Section 6.2.

6.2 Subdivision matrices and their JSR

Let S be a univariate, linear, stationary, uniform, m-ary subdivision scheme with mask $\mathbf{a}$ and compact support. To begin with, we assume that S is convergent with basic limit function φ. To simplify arguments, we assume $\underline{a} \le 0 \le \overline{a}$. Defining the bi-infinite linear operator A by $A_{u,v} := a_{u-mv}$, the refinement rule (6.1) reads

$$P^{k+1} = AP^k. \tag{6.4}$$

We will see that m finite submatrices of A contain all information about the smoothness of S. Their dimension $d \times d$ is given by the number of translated basic limit functions that have influence on $[0,1]$.
With φ being the basic limit function, the limit function corresponding to P^0 satisfies

$$S^\infty P^0 = \sum_{u\in\mathbb{Z}} P^0_u \varphi(\cdot - u) \tag{6.5}$$

due to uniformity and linearity of S.
When starting the subdivision process with initial data assigned to a grid $m^{-k}\mathbb{Z}$ instead of $\mathbb{Z}$, this leads to the basic limit function $\varphi(m^k\cdot)$. Since S is stationary, it is

$$S^\infty P^0 = S^\infty P^k = \sum_{u\in\mathbb{Z}} P^k_u \varphi(m^k \cdot - u). \tag{6.6}$$

Set $L := \{u : \operatorname{supp}(\varphi(\cdot - u)) \cap [0,1] \neq \emptyset\}$. Since S has compact support, φ has compact support and therefore L is a finite set. More precisely, $\operatorname{supp}\varphi(\cdot - u) = [u + \underline{a}, u + \overline{a}]$. Hence, $L = \{-\overline{a}+1, \dots, -\underline{a}\}$ and $d := \overline{a} - \underline{a}$ is the number of elements in L. By assumption, $0 \in L$.

Due to (6.5), the limit function corresponding to P^0 depends in the interval $[j, j+1]$ only on the values $\{P^0_u : u - j \in L\}$:

$$S^\infty P^0\big|_{[j,j+1]} = \sum_{u-j\in L} P^0_u \varphi(\cdot - u)$$

Further, (6.6) implies that $\{P_u^k : u - j \in L\}$ determines the limit function on $m^{-k}[j, j+1]$ by

$$S^\infty P^0\big|_{m^{-k}[j,j+1]} = \sum_{u-j\in L} P_u^k \varphi(m^k \cdot - u). \tag{6.7}$$

By linearity of the subdivision rules, we can define a family of matrices $\mathscr{A} = \{A_1, \dots, A_m\}$ that maps values of refinement step k to values of step $k+1$ such that, for $i = 1, \dots, m$,

$$A_i : \{P_u^k : u - j \in L\} \mapsto \{P_u^{k+1} : u - (mj + i - 1) \in L\}. \tag{6.8}$$

Then A_i maps values determining the limit function on $m^{-k}[j, j+1]$ to those determining it on $m^{-(k+1)}[mj + i - 1, mj + i] = m^{-k}[j + \frac{i-1}{m}, j + \frac{i}{m}]$. Due to uniformity of S, these matrices are independent of j. According to (6.4), A_i is the $d \times d$-submatrix of A which consists of the rows[1] $L + i - 1$ and the columns L, i.e., $(A_i)_{s,t} = A_{-\overline{a}+i+s-1,-\overline{a}+t}$. This leads to the explicit formula

$$(A_i)_{s,t} = a_{(m-1)\overline{a}-1+s-m\cdot t+i}. \tag{6.9}$$

Possibly, the matrices in $\mathscr{A}$ share one or more zero column. This implies that the family is reducible and the problem can be split into lower dimensional ones as described in Section 2.3.

So, with knowledge of $\{P_u^0 : u \in L\}$, we can describe an arbitrarily small interval of the limit function via a product of the matrices in $\mathscr{A}$. Let

$$\mathbf{P}^k_{m^{-k}[j,j+1]} := \left(P^k_{-\overline{a}+\ell+j}\right)_{\ell=1,\dots,d} \text{ for } \mathrm{k} \in \mathbb{N}_0, \mathrm{j} \in \mathbb{Z}.$$

That is, $\mathbf{P}^k_{m^{-k}[j,j+1]}$ determines $S^\infty P^0$ on an interval of length m^{-k} in the sense of (6.7), and (6.8) reads

$$\mathbf{P}^{k+1}_{m^{-(k+1)}[mj+i-1,mj+i]} = A_i \mathbf{P}^k_{m^{-k}[j,j+1]}. \tag{6.10}$$

Hence, infinite products of $\mathscr{A}$ lead to values of the limit function: For $x \in \mathbb{R}$, the value $S^\infty P^0(x)$ is related to the infinite product given by its m-adic expansion $x = \sum_{i=0}^\infty d_i m^{-i}$ with $d_0 \in \mathbb{Z}$, $d_i \in \{0, \dots, m-1\}$ for $i > 0$. By the principle of nested intervals, the product $A_I \mathbf{P}^0_{[d_0,d_0+1]}$ with $I = [d_1 + 1, d_2 + 1, \dots]$ describes $S^\infty P^0(x)$. This is illustrated for $m = 3$ and $x = \frac{11}{27}$ in Figure 6.1. Moreover, since we assumed that $0 \in L$,

$$P_u^k = \left(\mathbf{P}^k_{m^{-k}[u,u+1]}\right)_{\overline{a}} \quad \text{for } \mathrm{u} \in \mathbb{Z}. \tag{6.11}$$

To motivate the construction of the matrices, we assumed S to be convergent such that continuous limit functions and a non-trivial basic limit function exist. That enabled us to connect products of $\mathscr{A}$ to the limit functions of S. But the construction of the subdivision matrices does not depend on convergence of S. Given any symbol with finitely many non-zero coefficients, we can build matrices according to (6.9) such that (6.10) and (6.11) hold. That allows us to check if a scheme is contractive:

[1] summation with the index set element-wise

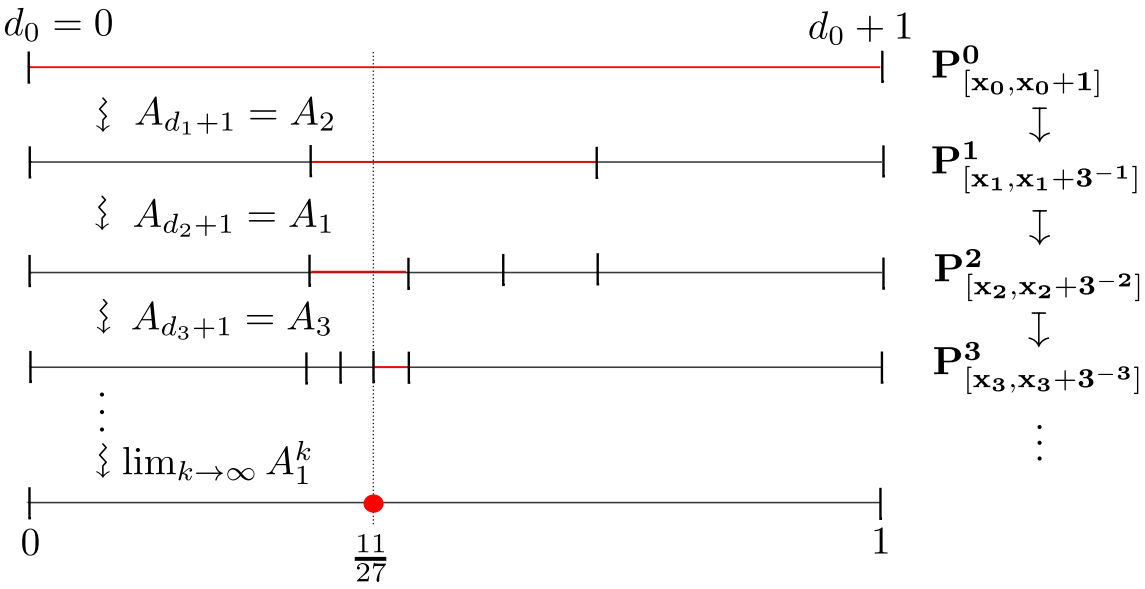


Figure 6.1: Illustration of the effect of $\mathscr{A}$ for $m = 3$ and $x = \frac{11}{27}$.

Theorem 6.4 *S is contractive if and only if $\hat{\rho}(\mathscr{A}) < 1$.*

Proof. We assume $\hat{\rho}(\mathscr{A}) < 1$. Then there exists $M \in \mathbb{N} \setminus \{0\}$ such that $\|A_I\|_\infty < 1$ for any $I \in \mathscr{I}_k, k \geq M$. Set $C := \max\{\|A_I\|_\infty : I \in \mathscr{I}_k, k < M\}$ and $\mu := \max\{\|A_I\|_\infty : I \in \mathscr{I}_M\} < 1$.
Consider $u \in \mathbb{Z}$ and $v = m^{-k}u$. Let P^0 be an arbitrary bounded initial sequence. Due to (6.11),

$$\left|P_u^k\right| \leq \left\|\mathbf{P}^k_{[v,v+m^{-k}]}\right\|_\infty = \left\|A_I \mathbf{P}^0_{[\lfloor v\rfloor,\lfloor v\rfloor+1]}\right\|_\infty$$

for some $I \in \mathscr{I}_k$ that is determined by the m-adic expansion up to order k of v. Splitting I into index vectors $I_i \in \mathscr{I}_M$ and a suffix S with $|S| < M$ such that $I = [I_1, \ldots, I_{\lfloor \frac{k}{M} \rfloor}, S]$, it follows that

$$\left|P_u^k\right| \leq \|S\|_\infty \left\|A_{I_{\lfloor \frac{k}{M} \rfloor}}\right\|_\infty \cdots \left\|A_{I_1}\right\|_\infty \left\|\mathbf{P}^0_{[\lfloor v\rfloor,\lfloor v\rfloor+1]}\right\|_\infty \leq C \cdot \mu^{\lfloor \frac{k}{M} \rfloor} \left\|P^0\right\|_\infty .$$

Regard $x \in \mathbb{R}$ with x_k being its m-adic expansion up to order k and $u := m^k x_k$. Due to the definition of $S^k P^0$,

$$\left|S^k P^0(x)\right| \leq \max\left(\left|S^k P^0\left(m^{-k}u\right)\right|, \left|S^k P^0\left(m^{-k}(u+1)\right)\right|\right) = \max\left(\left|P_u^k\right|, \left|P_{u+1}^k\right|\right) \to 0.$$

Hence, $S^\infty P^0 \equiv 0$ for all bounded initial sequences P^0, which means that S is contractive.

Let S be contractive and assume that $\hat{\rho}(\mathscr{A}) \geq 1$. According to [HMR09], there exists a sequence $I \in \mathscr{I}$ such that $\left\|A_{I(k)}\right\| \geq 1$ for all $k \in \mathbb{N}_0$, with $I(k)$ being the prefix of I with length k. Consider an initial sequence P^0 with values in {-1,1} such

that for every $c \in \{-1,1\}^d$, there exists $u \in \mathbb{Z}$ that satisfies $c = [P^0_{u-\overline{a}+1}, \dots, P^0_{u-\overline{a}+d}]$ or, reformulated, $c = \mathbf{P}^0_{[u,u+1]}$.
Let b_k be a row of $A_{I(k)}$ such that $\left\|A_{I(k)}\right\|_\infty = \left\|b_k\right\|_\infty$. Define $c_k \in \{-1,1\}^d$ such that the signs of c_k and b_k coincide entry-wise. With $c_k = \mathbf{P}^0_{[u_k,u_k+1]}$, there exists $v \in \mathbb{Z}$ such that

$$1 \le \left\|A_{I(k)}\right\|_\infty = \left\|A_{I(k)}c_k\right\|_\infty = \left\|A_{I(k)}\mathbf{P}^0_{[u_k,u_k+1]}\right\|_\infty = \left\|\mathbf{P}^k_{m^{-k}[v,v+1]}\right\|_\infty.$$

This implies $P^k_{-\overline{a}+\ell+v} = S^k P^0(m^{-k}(-\overline{a}+\ell+v)) \ge 1$ for some $1 \le \ell \le d$. But since S is contractive, there exists $k_0 \in \mathbb{N}$ such that $S^k P^0(x) < 1$ for any $x \in \mathbb{R}$ and $k \ge k_0$, leading to a contradiction. □

For smoothness information on S, the (divided) difference schemes must be considered: Let $\mathscr{A}$ be the matrix family deduced from $S_{\mathbf{b}}$ with symbol $b(z)$. According to Theorem 6.2, $S_{\mathbf{a}}$ with $a(z) = \frac{\left(\sum_{i=0}^{m-1} z^i\right)^{k+1}}{m^k} b(z)$ is C^k if $\hat{\rho}(\mathscr{A}) < 1$, the converse holds due to Theorem 6.3 if $S_{\mathbf{a}}$ is stable.
Hölder regularity is determined by the value of $\hat{\rho}(\mathscr{A})$, as Theorem 6.5 states. If $a(z)$ is divisible by $\left(\sum_{i=0}^{m-1} z^i\right)^\ell$ for $\ell = k$ but not for $\ell = k+1$, we call

$$c(z) = \frac{m^k}{\left(\sum_{i=0}^{m-1} z^i\right)^k} a(z)$$

the *kernel* of the scheme $S_{\mathbf{a}}$, and $\frac{c(z)}{m}$ the *scaled kernel*.

Theorem 6.5 *Let $a(z) = \frac{\left(\sum_{i=0}^{m-1} z^i\right)^{k+1}}{m^k} b(z)$ and $\mathscr{A}$ be deduced from $S_{\mathbf{b}}$.*

a) $S_{\mathbf{a}}$ is $C^{k+\alpha}_*$ with $\alpha = -\log_m(\hat{\rho}(\mathscr{A}))$. Moreover, $\alpha \le 1$.

b) If $S_{\mathbf{a}}$ is stable, the following holds: If either $\alpha = 1$ and $b(z)$ is the scaled kernel of $S_{\mathbf{a}}$ or $\alpha < 1$, the Hölder regularity $k+\alpha$ is maximal.

We remark that α is not necessarily positive such that we do not call it Hölder exponent of $S_{\mathbf{a}}$. The smoothness $C^{\lfloor k+\alpha \rfloor}$ and the Hölder exponent

$$\beta = k + \alpha - \lfloor k+\alpha \rfloor$$

arise from $k+\alpha = \lfloor k+\alpha \rfloor + \beta$.
Proof. In case $m = 2$, this is a direct consequence of [Rio92], Theorem 1.11. Translated to our notation, Rioul therein defines a Hölder regularity estimate $k+\alpha^j$ by

$$2^{-j\alpha^j} = \max_{I \in \mathscr{I}_j} \|A_I\|_\infty$$

and proves that $(\alpha^j)_j$ converges to $\alpha := \sup_j \alpha^j \leq 1$, and further, that $S_{\mathbf{a}}$ is $C_*^{k+\alpha}$. With

$$\alpha = \lim_{j\to\infty} \alpha^j = -\log_2 \left(\lim_{j\to\infty} \max_{I\in\mathscr{I}_j} \|A_I\|_\infty^{\frac{1}{j}} \right) = -\log_2(\hat{\rho}(\mathscr{A})),$$

it follows a). Furthermore, Rioul proves that $k+\alpha$ is maximal in the setting of b).

All arguments are analogous for arbitrary m, using theorems 6.2 and 6.3. □

6.3 Palindromic symmetry of binary schemes

Applications usually require subdivision schemes to be symmetric. Given a finite number of control points, it should not make any difference if a designer starts his input with the first or with the last one. In case of binary schemes, this leads to a special symmetry relation of the matrices in $\mathscr{A}$:

Definition 6.6 *A family $\mathscr{A} = \{A_1, A_2\}$ is called* palindromic *if $A_2 = RA_1R$ with R being the matrix with entries 1 on the counter-diagonal and 0 elsewhere:*

$$R = (r_{ij})_{i,j=1,\dots,d} \text{ with } r_{ij} = \begin{cases} 1 & i = d-j+1 \\ 0 & \text{else} \end{cases}.$$

Although our interest for palindromic families arises from analyzing symmetric subdivision schemes, the results of this section are not restricted to subdivision matrices but hold in general for palindromic families $\mathscr{A} = \{A_1, A_2\}$.
As discussed below, the palindromic symmetry of the matrices has consequences for the set-valued tree method. Choosing the norm adequately, it allows to significantly reduce the computational effort. But as another consequence, a palindromic family rarely possesses a spectral gap at 1. As we will see later on, families with generators of length 2 are an exception. To handle other cases, we present a transformation that, in certain situations, leaves the JSR unchanged but induces a spectral gap at 1 such that Theorem 3.15 becomes applicable.

The following notations may appear rather technical but allow a very compact display of results and proofs.

Definition 6.7 *For $i \in \mathscr{I}_1 = \{1,2\}$, we set $\bar{i} := \begin{cases} 2 & \text{if } i = 1 \\ 1 & \text{if } i = 2 \end{cases}$ and for $I = [i_1, \dots, i_\ell] \in \mathscr{I}_\ell, \ell \in \mathbb{N}_0$,*

$$\bar{I} := [\bar{i}_1, \dots, \bar{i}_\ell].$$

We observe that palindromic families induce a symmetric setting in terms of spectral radii: Due to $R = R^{-1}$, it is $A_I = RA_{\bar{I}}R^{-1}$ and consequently $\rho(A_I) = \rho(A_{\bar{I}})$. Therefore, we assume in the following that $J \in \mathscr{J}$ if and only if $\bar{J} \in \mathscr{J}$. That is, generators only come in pairs.

Against this background, Definition 6.7 can be generalized from completely positive to arbitrary index vectors:

Definition 6.8 *We call* $\{J_i, J_j\}, 1 \le i, j \le n$ *a* generator pair *of* $\mathscr{J} = \{J_1, \ldots, J_n\}$ *if* $J_i = \bar{J}_j$. *With* $\overline{-i} := -j$, *we denote for* $K = [k_1, \ldots, k_\ell] \in \mathscr{K}_\ell$

$$\bar{K} := [\bar{k}_1, \ldots, \bar{k}_\ell].$$

Obviously, $\bar{\bar{K}} = K$.

In case of choosing an adequate norm, the symmetric situation, which we observed in terms of spectral radii, is extended to a symmetry of the tree T in terms of node properties such that regarding half of the tree is sufficient. This is stated more formally by Proposition 6.10. The result is not difficult to prove but an important observation since it allows to reduce the complexity of the algorithms described in Chapter 4 to a half. To include norm $\|\cdot\|_{\text{poly}}$ as defined in Section 4.4 in the result, we need the following lemma.

Lemma 6.9 *Let* $\|\cdot\|_{\text{poly}}$ *be a norm whose unit ball is constructed by Algorithm 4.11 with start vectors* V_0 *for the iterated construction process. If* $v \in V_0 \Leftrightarrow Rv \in V_0$, *then* $\|RAR\|_{\text{poly}} = \|A\|_{\text{poly}}$.

Proof. Assume that $v \in V_0 \Leftrightarrow Rv \in V_0$. The following notations correspond to those in Algorithm 4.11.
By induction on k, we show that

$$v \in V_k \Rightarrow Rv \in V_k$$

holds for all iteration steps k. The converse holds due to $v = RRv$ trivially.
For $k = 0$, this is true by assumption. Let $v \in V_k$ and $Y_k := X_k \cup V_{k-1}$. We show that $Rv \in Y_k$:
Since $v \in Y_k$, either $v \in V_{k-1}$ or $v \in X_k$. In the first case, $Rv \in V_{k-1} \subseteq Y_k$ by induction hypothesis. In the second case, there exists $A_i \in \mathscr{A}$ and $\tilde{v} \in V_{k-1}$ such that $v = A_i\tilde{v}$. Therewith,

$$Rv = RA_i\tilde{v} = RA_iRR\tilde{v} = A_{\bar{i}}R\tilde{v}.$$

By induction hypothesis, $R\tilde{v} \in V_{k-1}$, implying $Rv \in X_k \subseteq Y_k$.
It remains to show that Rv is a vertex of $\text{conv}(Y_k)$. Assume that there is a proper convex combination

$$Rv = \sum_{y \in V_k} \lambda_y y.$$

Then

$$v = \sum_{y \in V_k} \lambda_y R y.$$

But we know that $Ry \in Y_k$ since $y \in V_k \subset Y_k$. Hence, v is proper convex combination of elements of Y_k which contradicts the property of v to be a vertex of $\mathrm{conv}(Y_k)$. Therewith, Rv is an element of V_k.
This implies that $v \in \mathcal{V}$ if and only if $Rv \in \mathcal{V}$. Furthermore, $n^T v = 1$ if and only if $(n^T R)(Rv) = 1$. If $\{x : n^T x \leq 1\}$ is non-redundant for the description of $\mathrm{conv}(\mathcal{V})$, then so is $\{x : n^T R x \leq 1\}$. Hence, by construction, n^T is a row of N if and only if $n^T R$ is a row of N. So, up to permutation of rows and columns, $NRARV$ equals NAV and therefore $\|A\|_{\mathrm{poly}} = \max_{i,j}(NAV)_{ij} = \max_{i,j}(NRARV)_{ij} = \|RAR\|_{\mathrm{poly}}$. □

Proposition 6.10 *Let $\mathcal{A}$ be a palindromic family with generator set $\mathcal{J}$, consider as norm $\|\cdot\|_1$, $\|\cdot\|_2$, $\|\cdot\|_\infty$ or $\|\cdot\|_{\mathrm{poly}}$ satisfying the condition of Lemma 6.9.*

a) If there exists a $\mathcal{J}$-complete tree descending from 1, then $\hat{\rho}(\mathcal{A}) = 1$.

b) If there exists a contractive tree descending from 1, then $\hat{\rho}(\mathcal{A}) < 1$.

Proof. The proof is done in the following steps: First, we show that $\|\mathcal{A}_K\| = \|\mathcal{A}_{\bar{K}}\|$ for the specified norms. That implies that K is (strictly) 1-bounded if and only if so is $\bar{K}$. Second, we show that K is covered if and only if so is $\bar{K}$. Third, we construct a $\mathcal{J}$-complete resp. contractive tree descending from $\emptyset$, which shows a) resp. b).

An important relation of $K \in \mathcal{K}$ and $\bar{K}$ is given by

$$A_I \in \mathcal{A}_K \Leftrightarrow RA_IR \in \mathcal{A}_{\bar{K}}, \quad I \in \mathcal{I}. \tag{6.12}$$

This is obvious for K being completely positive since $R = R^{-1}$. Assume that K has exactly one negative entry, $K = [P, -i, S]$ with $i > 0$. Then

$$\begin{aligned}\mathcal{A}_K &= \{A_S A_{J_i}^k A_P : k \in \mathbb{N}_0\} = \{RA_{\bar{S}} R (RA_{\bar{J}_i} R)^k RA_{\bar{P}} R : k \in \mathbb{N}_0\} \\ &= \{RA_{\bar{S}} A_{\bar{J}_i}^k A_{\bar{P}} R : k \in \mathbb{N}_0\} = \{RAR : A \in \mathcal{A}_{\bar{K}}.\}\end{aligned}$$

(6.12) follows by induction on the number of negative entries.

For the specified norms, $\|A_{\bar{I}}\| = \|A_I\|$ for $I \in \mathcal{I}$: This corresponds to Lemma 6.9 for $\|\cdot\|_{\mathrm{poly}}$ satisfying the condition $v \in V_0 \iff Rv \in V_0$. Since R is unitary, the equality holds for the unitarily invariant norm $\|\cdot\|_2$. The norms $\|\cdot\|_\infty$ and $\|\cdot\|_1$ are invariant under permutation of rows and columns.
It follows from (6.12) that $\|\mathcal{A}_K\| = \|\mathcal{A}_{\bar{K}}\|$ for $K \in \mathcal{K}$.

Consider a covered node K, i.e., $K = [P, S]$ with $\mathcal{A}_K \in \mathcal{A}_P$ and S completely positive. Then also $\bar{K} = [\bar{P}, \bar{S}]$ is covered: Since S is completely positive, $\bar{S}$ is as well, and $A \in \mathcal{A}_{\bar{K}} \Rightarrow RAR \in \mathcal{A}_K \Rightarrow RAR \in \mathcal{A}_P \Rightarrow A \in \mathcal{A}_{\bar{P}}$.

Let T_* be a $\mathscr{J}$-complete resp. contractive tree descending from 1, $\mathscr{K}_*$ the set of its nodes and $\bar{\mathscr{K}}_* := \{\bar{K} : K \in \mathscr{K}_*\}$. By definition of $\bar{k}$ for $k \in \mathscr{K}_1$, $[1, I] \in \mathscr{K}_*$ has the same number of positive or negative children as $[2, \bar{I}] \in \bar{\mathscr{K}}_*$. In particular, $\bar{L}$ is a leaf of $\bar{T}_*$ if and only if L is a leaf of T_* and therefore (strictly) 1-bounded or covered. Hence, $\bar{T}_*$ is J-complete resp. contractive descending from 2. This obviously implies that the tree with nodes $\mathscr{K}_* \cup \bar{\mathscr{K}}_* \cup \emptyset$ is $\mathscr{J}$-complete resp. contractive. Then a) follows with Theorem 3.6 resp. b) with Lemma 3.5. □

While the set-valued tree method benefits from the symmetry of norms, the symmetry of spectral radii is a disadvantage: A palindromic family $\mathscr{A}$ admits a spectral gap at 1 only if $\bar{J} \in \Pi(J)$. If $|J| = 2$, this is always the case: If $[1,1]$ resp. $[2,2]$ is a strong generator, then $[1]$ resp. $[2]$ is a strong generator as well. Hence, $[1,1]$ resp. $[2,2]$ is no candidate for a dominant generator. The situation that remains to consider is $J = [1,2]$, and $\bar{J} = [2,1] \in \Pi(J)$.
But in general, appearance of a spectral gap is particular. Instead of a dominant generator, we expect $\mathscr{A}$ to admit a dominant pair, sharing a spectral gap as we define here:

Definition 6.11 *The matrix family $\mathscr{A}$ has a* shared spectral gap at 1 *if there exists a generator set $\mathscr{J} = \{J, \bar{J}\}$ such that*

- *there exists $q < 1$ such that*

$$\rho(A_I) \leq q \tag{6.13}$$

  for any product A_I, unless $I = \emptyset$ or $I = [S, J^r, P]$ for some $r \in \mathbb{N}_0$ and some partition $[P,S] = J$ of J, or $I = [S, \bar{J}^r, P]$ for some $r \in \mathbb{N}_0$ and some partition $[P,S] = \bar{J}$ of $\bar{J}$.

- *the Jordan normal form Λ of A_J is*

$$\Lambda := V^{-1} A_J V = \begin{bmatrix} 1 & 0 \\ 0 & \Lambda_* \end{bmatrix}, \quad \rho(\Lambda_*) < 1. \tag{6.14}$$

In this case, $\{J, \bar{J}\}$ is called a dominant pair.

Theorem 3.15 guaranteed the existence of a $\mathscr{J}$-complete tree in case of a spectral gap at 1, but not in case of a *shared* spectral gap at 1. Hence, most palindromic families will not satisfy the conditions of Theorem 3.15. In order to save the result for the palindromic situation, we introduce a transformed family.

Definition 6.12 *If $\mathscr{A} = \{A_1, A_2\}$ is palindromic, we call the family $\mathscr{E} = \{E_1, E_2\}$ with $E_1 := A_1 R$ and $E_2 := R A_1$ the* palindromic transform *of $\mathscr{A}$.*

We will see that $\mathscr{E}$ is palindromic as well, see Lemma 6.16. Furthermore, proving $\hat{\rho}(\mathscr{A}) = 1$ is equivalent to proving $\hat{\rho}(\mathscr{E}) = 1$:

Lemma 6.13 *If $\mathscr{E}$ is the palindromic transform of $\mathscr{A}$, then*

$$\hat{\rho}(\mathscr{A}) = \hat{\rho}(\mathscr{E}).$$

Proof. It is easy to verify that $\{E_I | I \in \mathscr{I}_2\} = \{A_I | I \in \mathscr{I}_2\}$ and so obviously

$$\{E_I | I \in \mathscr{I}_{2k}\} = \{A_I | I \in \mathscr{I}_{2k}\}$$

for any $k \in \mathbb{N}$. Therewith,

$$\begin{aligned}
\hat{\rho}(\mathscr{E}) &= \lim_{k\to\infty} \max_{I\in\mathscr{I}_k} \|E_I\|^{\frac{1}{k}} = \lim_{k\to\infty} \max_{I\in\mathscr{I}_{2k}} \|E_I\|^{\frac{1}{2k}} \\
&= \lim_{k\to\infty} \max_{I\in\mathscr{I}_{2k}} \|A_I\|^{\frac{1}{2k}} = \lim_{k\to\infty} \max_{I\in\mathscr{I}_k} \|A_I\|^{\frac{1}{k}} \\
&= \hat{\rho}(\mathscr{A}) = 1.
\end{aligned}$$

□

For a compact description of the relation between $\mathscr{A}$ and $\mathscr{E}$, we establish the following notation:

Definition 6.14 *For $I = [i_1, \ldots, i_\ell] \in \mathscr{I}_\ell$, define $\underline{I}$ by*

$$(\underline{I})_j := \begin{cases} \bar{i}_j & \text{if } j \text{ odd} \\ i_j & \text{if } j \text{ even} \end{cases}$$

for $j = 1, \ldots, \ell$.

Obviously, $\underline{\underline{I}} = I$.

Theorem 6.15 exhibits the benefit of the palindromic transformation.

Theorem 6.15 *Consider a palindromic family $\mathscr{A}$ that is product bounded and has a shared special gap at 1 with dominant pair $\{J, \bar{J}\}$. If $|J|$ is odd, then $\mathscr{E}$ is product bounded and has a special gap at 1 so that $\mathscr{E}$ satisfies the conditions of Theorem 3.15. In that case, $\hat{J} := [\underline{J}, \underline{\bar{J}}]$ is a dominant generator of $\mathscr{E}$.*

In order to prove Theorem 6.15, the following lemma links the products of $\mathscr{E}$ with those of $\mathscr{A}$.

Lemma 6.16

a) $E_i = A_i R = R A_{\bar{i}}$ *for* $i = 1, 2$.

b) $E_I = A_{\underline{I}}$ *for* $I \in \mathscr{I}$ *with* $|I|$ *even.*

c) $E_I = R A_{\underline{I}}$ *for* $I \in \mathscr{I}$ *with* $|I|$ *odd.*

d) $E_{\bar{I}} = R E_I R$ *for any* $I \in \mathscr{I}$. *In particular,* $\mathscr{E}$ *is palindromic.*

e) $E_{[\underline{I},\bar{I}]} = A_{I^2}$ *for* $I \in \mathscr{I}$ *with* $|I|$ *odd.*

Proof.

a) The claim follows from (6.12).

b) This is easy to verify for E_I with $|I| = 2$. Regard a product of length $2k$ as product of length k with factors being products of length 2. Then use the fact that for $K, L \in \mathscr{I}$ with K, L even, $[\underline{K}, \underline{L}] = \underline{[K, L]}$.

c) $|I|$ is odd, hence $I = [P, i]$ with $|P|$ even and $i \in \{1, 2\}$. It follows with a) and b) that $E_I = E_i E_P = R A_{\bar{i}} A_{\underline{P}} = R A_{[\underline{P},\bar{i}]} = R A_{\underline{I}}$

d) The claim follows trivially for $I \in \mathscr{I}_1$. For $I = \{i_1, \ldots, i_k\} \in \mathscr{I}_k$, $R E_I R = R E_{i_k} R \cdots R E_{i_2} R R E_{i_1} R = E_{\bar{I}}$.

e) This is a consequence of c).

□

Now, we are able to prove Theorem 6.15:
Proof. It is obvious that $\mathscr{E}$ is product bounded if $\mathscr{A}$ is product bounded: Due to Lemma 6.16 a) and b), $\|E_I\| \leq \max\{\|R\| \cdot \|A_{\underline{I}}\|, \|A_{\underline{I}}\|\}$.

We show that $\mathscr{E}$ has a spectral gap at 1. By Lemma 6.13, $\hat{\rho}(\mathscr{E}) = 1$. $\{\hat{J}, \bar{\hat{J}}\}$ is a generator set since $\rho(E_{\hat{J}}) = \rho(A_{J^2}) = \rho(A_J^2) = \rho(A_J)^2 = 1$ due to Lemma 6.16 e). Furthermore, $\bar{\hat{J}} = [\underline{\bar{J}}, \underline{J}]$ is a cyclic permutation of $\hat{J}$.
To see that $\hat{J}$ is dominant, we regard the products of $\mathscr{E}$, distinguishing products of even and odd length.
Considering products of even length, recall that $\{E_I : I \in \mathscr{I}_{2k}\} = \{A_I : I \in \mathscr{I}_{2k}\}$. With $q < 1$ from (6.13), we know by dominance of $\{J, \bar{J}\}$ that $\{A_I : I \in \mathscr{I}_{2k}\}$ contains at most $2 \cdot |J|$ products A with $\rho(A) \geq q$, namely the matrices according to even powers of $|J|, |\bar{J}|$ and their cyclic permutations. So there are at most $|\hat{J}| = 2 \cdot |J|$ products E with $\rho(E) \geq q$. Since $|\hat{J}|$ is the number of cyclic permutations $\pi(\hat{J})$ and $\rho(E_J) = \rho(E_{\pi(J)})$, $\rho(E_I) < q$ for $I \notin \Pi(\hat{J})$ even.
Considering products of odd length, assume that there is E_I with $I \in \mathscr{I}_{2k+1}$ such

that $\rho(E_I) \geq \sqrt{q}$. Then $\rho(E_{I^2}) \geq q$.
Furthermore, $E_I = RA_{\underline{I}}$ by Lemma 6.16 c). Therewith,

$$E_{I^2} = RA_{\underline{I}}RA_{\underline{I}} = A_{\bar{\underline{I}}}A_{\underline{I}} = A_{[\underline{I},\bar{\underline{I}}]}.$$

Due to dominance of $\{J, \bar{J}\}$ for $\mathscr{A}$, $[\underline{I}, \bar{\underline{I}}] = \pi(J)^\ell$ or $[\underline{I}, \bar{\underline{I}}] = \pi(\bar{J})^\ell$ with cyclic permutation π and $\ell \in \mathbb{N}$.
So, $2 \cdot |I| = \ell \cdot |J|$. Since $|J|$ odd, $\ell = 2 \cdot \tilde{\ell}$ for some $\tilde{\ell}$. Hence,

$$[\underline{I}, \bar{\underline{I}}] = [\Pi(J)^{\tilde{\ell}}, \Pi(J)^{\tilde{\ell}}]$$

or

$$[\underline{I}, \bar{\underline{I}}] = [\Pi(\bar{J})^{\tilde{\ell}}, \Pi(\bar{J})^{\tilde{\ell}}].$$

But this, together with $|\underline{I}| = |\bar{\underline{I}}|$, implies $\underline{I} = \bar{\underline{I}}$, which is impossible. It follows that $\rho(E_I) < \sqrt{q}$ for all $I \in \mathscr{I}_{2k+1}$.
Summarizing, for any product E_I with I not being power of $\hat{J}$ or of its cyclic permutations, we showed $\rho(E_I) < \sqrt{q}$.

With $\Lambda_{\hat{J}}$ resp. Λ_J being the Jordan normal form of $E_{\hat{J}}$ resp. A_J, it is

$$\Lambda_{\hat{J}} = V^{-1}E_{\hat{J}}V = V^{-1}A_J^2V = (\Lambda_J)^2 = \begin{bmatrix} 1 & 0 \\ 0 & \Lambda_*^2 \end{bmatrix}.$$

Hence, $\hat{J}$ is dominant for $\mathscr{E}$. □

Summarized, the symmetrical situation of palindromic families can be utilized to reduce the complexity of the problem as specified in Proposition 6.10. But the symmetrical appearance of generators prohibits in general the desirable situation of a spectral gap at 1. There is no problem in the quite typical case of generator length 2, occurring systematically for palindromic 2×2-matrices, see [Möß10]. The palindromic transform allows to handle the cases with generators of odd length. For the transformed family, Theorem 6.15 guarantees the existence of a $\mathscr{J}$-complete tree under certain conditions.
This result is theoretically important. In practice, the palindromic transformation does not necessarily lead to shorter or trimmer trees as some of the examples in Chapter 7 demonstrate. Note also that the norm $\|\cdot\|_V$ resp. $\|\cdot\|_w$ as defined in Section 3.3 resp. Section 4.4 destroys the symmetric situation. Therefore, Proposition 6.10 cannot be applied to $\mathscr{E}$ when searching for the tree whose existence is guaranteed.

7 Examples

The examples presented in this chapter serve several purposes. They shall demonstrate the capabilities of the method presented in this work. This was the aim of the current implementation of Algorithm 4.3, which is not optimized in terms of runtime. Furthermore, they illustrate the algorithmic details discussed in Chapter 4, as the use of different parameter values or different choices of norm. Section 7.1 presents several elementary examples, illustrating the method for families with spectral gap at 1 at well as for families with more than one strong generator. Most of the other examples result from smoothness analysis of subdivision schemes, providing the exact Hölder regularity of these schemes. Families with $m > 2$ are represented by ternary and quaternary schemes. As benchmarks for the implemented algorithm serve the parameter depending families of the primal and dual 4-point scheme.
Algorithm 4.12 was applied to answer the open question if parameter values exist for which the parametrized 8-point scheme is C^4.
The trees in Section 7.1 are calculated analytically with use of MAPLE. All other examples are computed in MATLAB, and a node K is recognized to be 1-bounded if and only if $\|N_K\| < 1 - 10^{-7}$ is numerically true. Unless indicated otherwise, the check for 1-boundedness bases on the approach using balls of matrices as described in Section 4.3.1.

Recall that the input $\mathscr{A}$ of Algorithm 4.3 is a family for which $\hat{\rho}(\mathscr{A}) = 1$ is to be validated. Therefore, a family $\mathscr{A}$ resulting from applications has to be scaled as explained in Section 2.4. Please note that all $\mathscr{J}$-complete trees displayed in the following correspond to the scaled family $\rho(A_{J_1})^{-\frac{1}{|J_1|}}\mathscr{A}$, implying[1] that J_1 is a strong generator of the generator set $\mathscr{J} = \{J_1, \ldots, J_n\}$.

Clearly, the numerical results in this chapter cannot be taken as rigorous mathematical proves. The numerically computed $\mathscr{J}$-complete tree is to be taken as a strong indication that the JSR of the exact scaled family equals 1. In Section 3.3, stability under sufficiently small perturbations is discussed. In principle, the nature of the problem allows a treatment via interval arithmetic. When scaling the matrix family in such a framework, it is important to ensure that the leading eigenvalues of the generator matrix are not described by an enclosing interval but, since their values are known by construction, are the precise values with modulus 1. Otherwise, the special behavior of strong generator powers, which neither tend to

[1] Otherwise, $\mathscr{J}$ would not be a generator set for the scaled family.

infinity nor to zero, is not modeled. In fact, also the MATLAB implementation makes use of the knowledge of the leading eigenvalue of a strong generator, see Section 4.3.3.

A few of the following examples were also published in [MR14] for illustrative purposes.

7.1 Illustrating Examples

In this section, we illustrate aspects of our method by considering some model problems. Unless indicated otherwise, all trees are constructed with respect to the maximum absolute row sum norm $\|\cdot\|_\infty$.

First, let $\mathscr{A} = \{A_1, A_2\}$ with

$$A_1 = \begin{bmatrix} \frac{10}{9} & \frac{1}{3} \\ -\frac{1}{3} & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & \frac{1}{5}\sqrt{1-\varepsilon} \\ -\frac{1}{5}\sqrt{1-\varepsilon} & \frac{26}{25} - \varepsilon \end{bmatrix}.$$

Then $\rho(A_1) = 1$ and $\rho(A_2) = 1 - \varepsilon$. Let us consider a few special cases:

- For $\varepsilon = \frac{1}{8}$, we choose the generator set $\mathscr{J} = \{[1]\}$. It leads to a $\mathscr{J}$-complete tree, i.e., the conditions of Theorem 3.6 are satisfied, see Figure 7.1 *(left)*. Therewith, $\hat{\rho}(\mathscr{A}) = \rho(A_1) = 1$.

- For $\varepsilon = 0$, both matrices A_1, A_2 have spectral radius 1. Hence, $\mathscr{A}$ does *not* have a spectral gap at 1. Figure 7.1 *(right)* shows the $\mathscr{J}$-complete tree for $\mathscr{J} = \{J_1, J_2\}$ with $J_1 = [1], J_2 = [2]$, thus proving $\hat{\rho}(\mathscr{A}) = 1$ also in this case.

- For ε close to 0, the asserted benefits of generators J with $\rho(A_J) < 1$ become apparent. For $\varepsilon = 0.01$, the spectral radius of A_2 is 0.99. In principle, it is sufficient to use only the generator $J_1 = [1]$, see Figure 7.2 *(left)*. However, Figure 7.2 *(right)* shows that the depth of the resulting tree is reduced significantly when using the additional weak generator $J_2 = [2]$. This is because the slow decay of norms of matrix powers $\|A_2^k\|$ is subsumed in the single negative node marked by a triangle. For smaller values of ε, the effect becomes even more drastic. For instance, $\varepsilon = 0.001$ leads to depth 224 when using a single generator, while the two generators still yield the same trim pattern shown in Figure 7.2 *(right)*.

The next two examples are taken from [GWZ05]. These pairs of matrices are not asymptotically simple, prohibiting a validation by means of a finite structure when following the approach suggested there. In contrast, our method meets the challenge and establishes the JSR in a rather efficient way.

Let $\mathscr{B} = \{B_1, B_2\}$ with

$$B_1 = \begin{pmatrix} \cos(1) & -\sin(1) \\ \sin(1) & \cos(1) \end{pmatrix}, \quad B_2 = \begin{pmatrix} \frac{1}{2} & -\frac{i}{2} \\ 0 & 0 \end{pmatrix}.$$

When choosing $\mathscr{J} = \{[1]\}$, we obtain a $\mathscr{J}$-complete tree, see Figure 7.3 *(left)*. Thus, $\hat{\rho}(\mathscr{B}) = \rho(B_1) = 1$.

Let $\mathscr{C} = \{C_1, C_2\}$ with

$$C_1 = \frac{3-\sqrt{5}}{2}\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}, \quad C_2 = \frac{3-\sqrt{5}}{2}\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}.$$

With $\mathscr{J} = \{[1], [2]\}$, we obtain a $\mathscr{J}$-complete tree, see Figure 7.3 *(right)*. Thus, $\hat{\rho}(\mathscr{C}) = \rho(C_1) = \rho(C_2) = 1$.

Let $\mathscr{D} = \{D_1, D_2\}$ with

$$D_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad D_2 = \frac{9}{10}\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

In [GP13], this example illustrates the method, and the unit ball of the extremal norm was computed in two iterations. With $\mathscr{J} = \{[1,2]\}$, Algorithm 4.3 terminates with the $\mathscr{J}$-complete trees of depth 8 visualized in Figure 7.4 in a subsecond. This confirms $\hat{\rho}(\mathscr{D}) = \rho(D_2D_1)^{\frac{1}{2}}$. The tree on the *left* corresponds to $\|\cdot\|_2$, the one on the *right* to $\|\cdot\|_{\text{poly}}$ with $\text{IT} = 3$. The parameter values for the computation were in both cases set to $\text{LIMRADCONSTR} = \frac{1}{100}$ and $\text{LIMRADCOMPUT} = \frac{1}{17}$. With these parameter values and $\|\cdot\|_\infty$, the algorithm terminates in $\text{MAXLEVEL} = 50$ with output "No result". Comparison of the trees indicates that $\|\cdot\|_{\text{poly}}$ is, at least for small IT, not necessarily a better choice than the non-adapted norm $\|\cdot\|_2$.
Furthermore, the value of LIMRADCOMPUT is important here. When changing LIMRADCOMPUT to $\frac{1}{16}$, the algorithm also terminates for the norms $\|\cdot\|_2$ and $\|\cdot\|_{\text{poly}}$ in $\text{MAXLEVEL} = 50$ with output "No result".

The set-valued tree method in principle abandons conditions on the matrix family as irreducibility or product boundedness. Although transforming the problem to the analysis of lower-dimensional irreducible families is certainly to recommend, it is pleasing that no a priori check for irreducibility has to be performed. The following simple example demonstrates successful termination of Algorithm 4.3 for a reducible family. Consider $\mathscr{F} = \{F_1, F_2\}$ with

$$F_1 = \begin{pmatrix} \frac{1}{4} & 0 \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}, \quad F_2 = \begin{pmatrix} 1 & -\frac{1}{4} \\ 0 & \frac{1}{2} \end{pmatrix}.$$

$\mathscr{F}$ is product bounded but not irreducible since F_1 and F_2 share the eigenvector $(1,2)^T$. With $\mathscr{J} = \{[2]\}$, the family possesses the $\mathscr{J}$-complete tree visualized in Figure 7.5 (*left*), validating $\hat{\rho}(\mathscr{F}) = \rho(F_2)$.

In case of a family which is not product bounded, the approach works if the unbounded products can be subsumed in negatives nodes. Consider $\mathscr{G} = \{G_1, G_2\}$ with

$$G_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad G_2 = \begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}.$$

$(G_1^k)_k$ is unbounded. But since $G_2 \cdot G_1^k = G_2$, a $\mathscr{J}$-complete tree shown in Figure 7.5 (*right*) exists for $\mathscr{J} = \{[1]\}$.

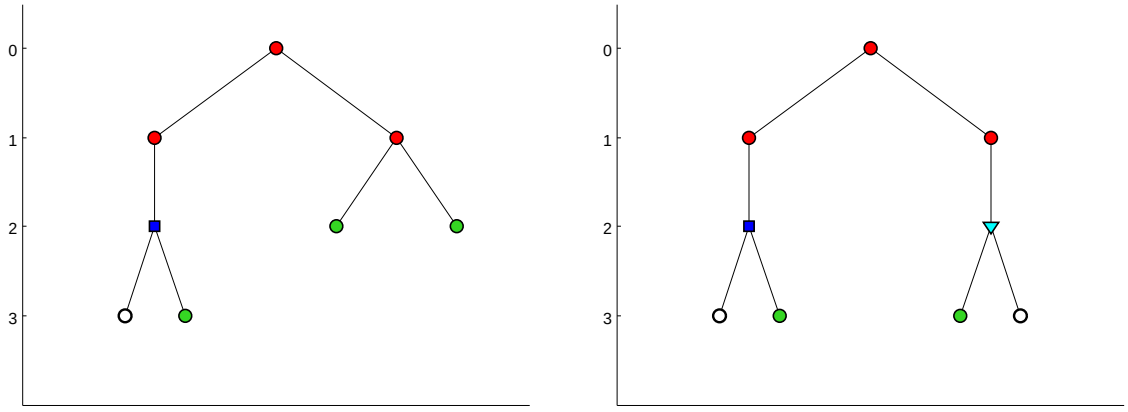


Figure 7.1: $\mathscr{J}$-complete trees for $\mathscr{A}$ with $\varepsilon = 1/8$ *(left)* and $\varepsilon = 0$ *(right)*, and generator set $\mathscr{J} = \{[1], [2]\}$. Colors and shapes of markers indicate properties of nodes: green $\triangleq$ 1-bounded, white $\triangleq$ covered, square $\triangleq$ negative child w.r.t. generator $J_1 = [1]$, triangle $\triangleq$ negative child w.r.t. generator $J_2 = [2]$, red $\triangleq$ other.

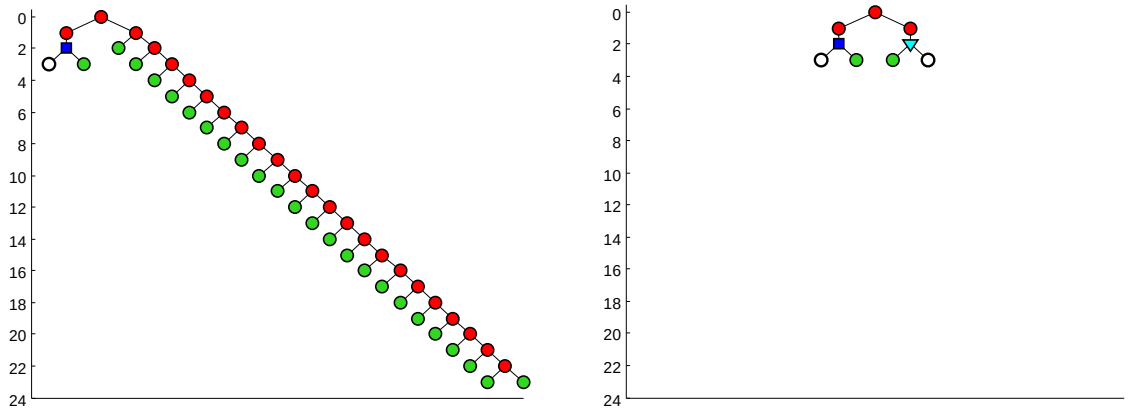


Figure 7.2: $\mathscr{J}$-complete trees for $\mathscr{A}$ with $\varepsilon = 0.01$, and generator set $\mathscr{J} = \{[1]\}$ *(left)* respective $\mathscr{J} = \{[1], [2]\}$ *(right)*.

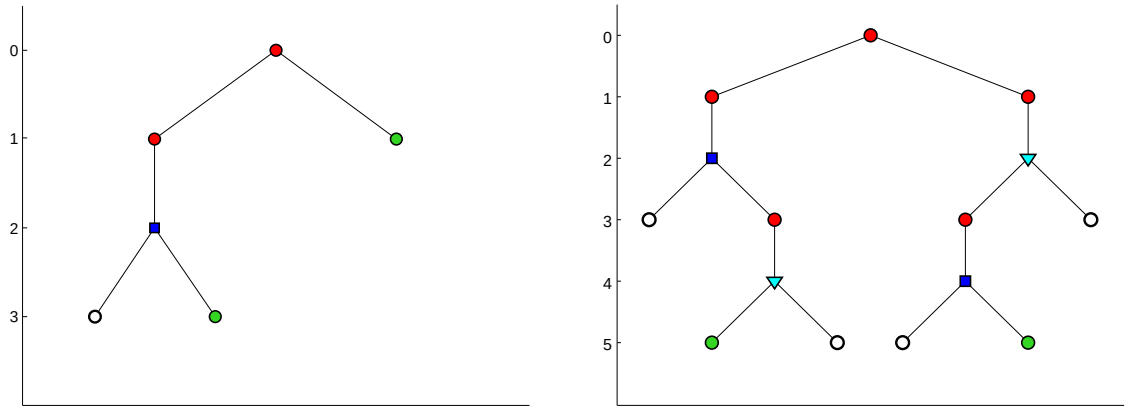


Figure 7.3: $\mathscr{J}$-complete trees for $\mathscr{B}$ with $\mathscr{J} = \{[1]\}$ *(left)* and for $\mathscr{C}$ with $\mathscr{J} = \{[1], [2]\}$ *(right)*.

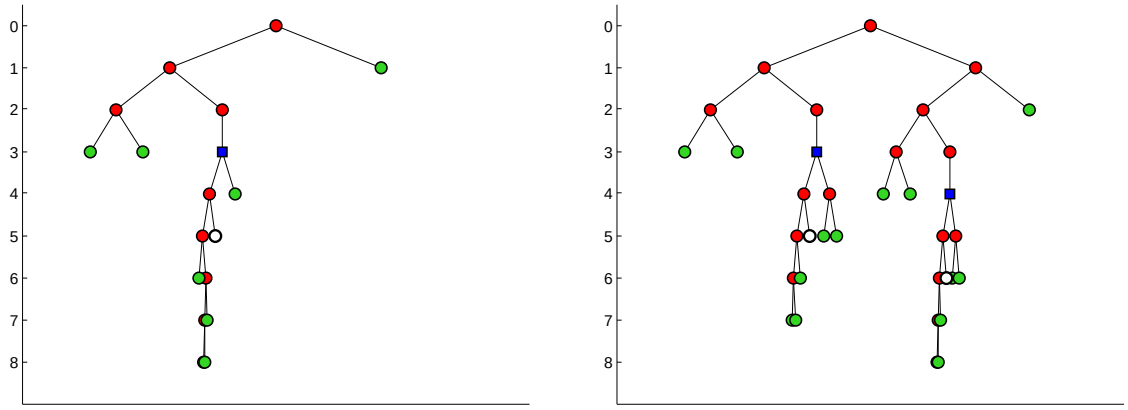


Figure 7.4: $\mathscr{J}$-complete trees for $\mathscr{D}$ with $\mathscr{J} = \{[1,2]\}$ and Norm $\|\cdot\|_2$ *(left)* respective $\|\cdot\|_{\text{poly}}$ with $\text{IT} = 3$ *(right)*.

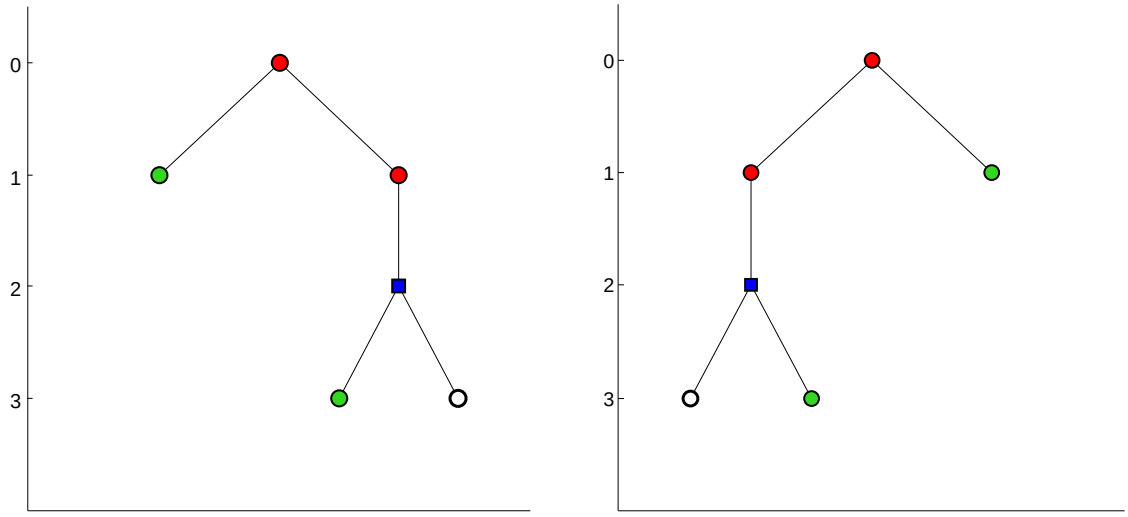


Figure 7.5: $\mathscr{J}$-complete trees for $\mathscr{F}$ with $\mathscr{J} = \{[2]\}$ *(left)*, and $\mathscr{G}$ with $\mathscr{J} = \{[1]\}$ *(right)*.

7.2 3-point scheme

Consider the binary 3-point scheme that is given by the symbol

$$a(z) = \frac{1}{32}\left(-3(z^{-2}+z^3)+5(z^{-1}+z^2)+30(1+z^1)\right).$$

Figure 7.6 illustrates how old points are mapped to new points. Checking the divided difference schemes with symbols $\frac{2}{(1+z)^2}a(z)$ and $\frac{4}{(1+z)^3}a(z)$ for contractivity shows that the original scheme is C^1 but not C^2. To compute Hölder regularity, we therefore have to analyze a scheme with the symbol $\frac{2^{k-1}}{(1+z)^k}a(z)$ and $k \geq 2$. For $k = 2$, the corresponding matrix family $\mathscr{A} = \{A_1, A_2\}$ is given by

$$A_1 = \frac{1}{32}\begin{pmatrix} 22 & -6 & 0 \\ -6 & 22 & 0 \\ 0 & 22 & -6 \end{pmatrix}, \quad A_2 = \frac{1}{32}\begin{pmatrix} -6 & 22 & 0 \\ 0 & 22 & -6 \\ 0 & -6 & 22 \end{pmatrix}.$$

For $k = 3$, the corresponding matrix family $\mathscr{B} = \{B_1, B_2\}$ is given by

$$B_1 = \frac{1}{32}\begin{bmatrix} 56 & 0 \\ -12 & -12 \end{bmatrix}, \quad B_2 = \frac{1}{32}\begin{bmatrix} -12 & -12 \\ 0 & 56 \end{bmatrix}.$$

The 3-point scheme has maximal Hölder regularity $\beta_* = 1+\alpha_* = 1-\log_2(\hat{\rho}(\mathscr{A})) = 2-\log_2(\hat{\rho}(\mathscr{B}))$. The set-valued tree approach is successful for both families $\mathscr{A}$ and $\mathscr{B}$ with $\mathscr{J} = \{[1],[2]\}$. Figure 7.7 presents $\mathscr{J}$-complete trees that are obtained for different choices of parameters values. The runtime is on a sub-second timescale. Any of the trees displayed in Figure 7.7 is $\mathscr{J}$-complete and therewith satisfies the condition of Theorem 3.6. We conclude $\hat{\rho}(\mathscr{A}) = \rho(A_1) = \rho(A_2) = \frac{7}{8}$ and $\beta_* = 4 - \log_2(7) \approx 1.1927$.

With $\|\cdot\|_2$, LimRadConstr $= 1/100$ and LimRadComput $= 1/10$, a $\mathscr{J}$-complete tree of $\mathscr{B}$ is computed whose visualization is equivalent to the tree 1a in Figure 7.7. Hence, $\hat{\rho}(\mathscr{B}) = \rho(B_1) = \rho(B_2) = \frac{7}{4}$, confirming the result for β_*. Additionally, this accords with the explicit formula of [Möß10] for palindromic (2×2)-matrices.

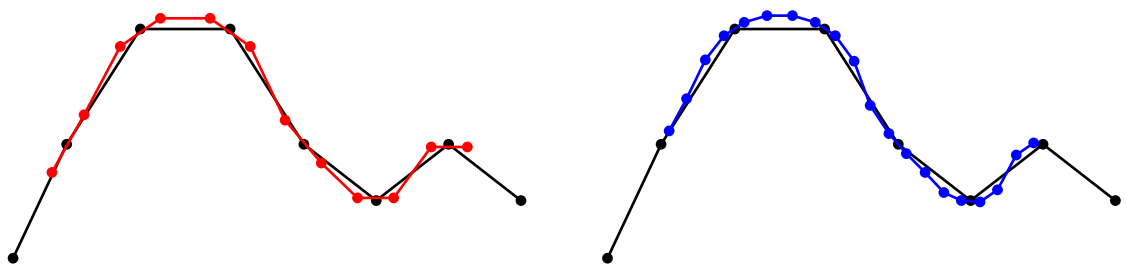


Figure 7.6: Visualization of the 3-point scheme, showing the control points after the first (red) and second (blue) subdivision step in comparison to the initial control points (black).

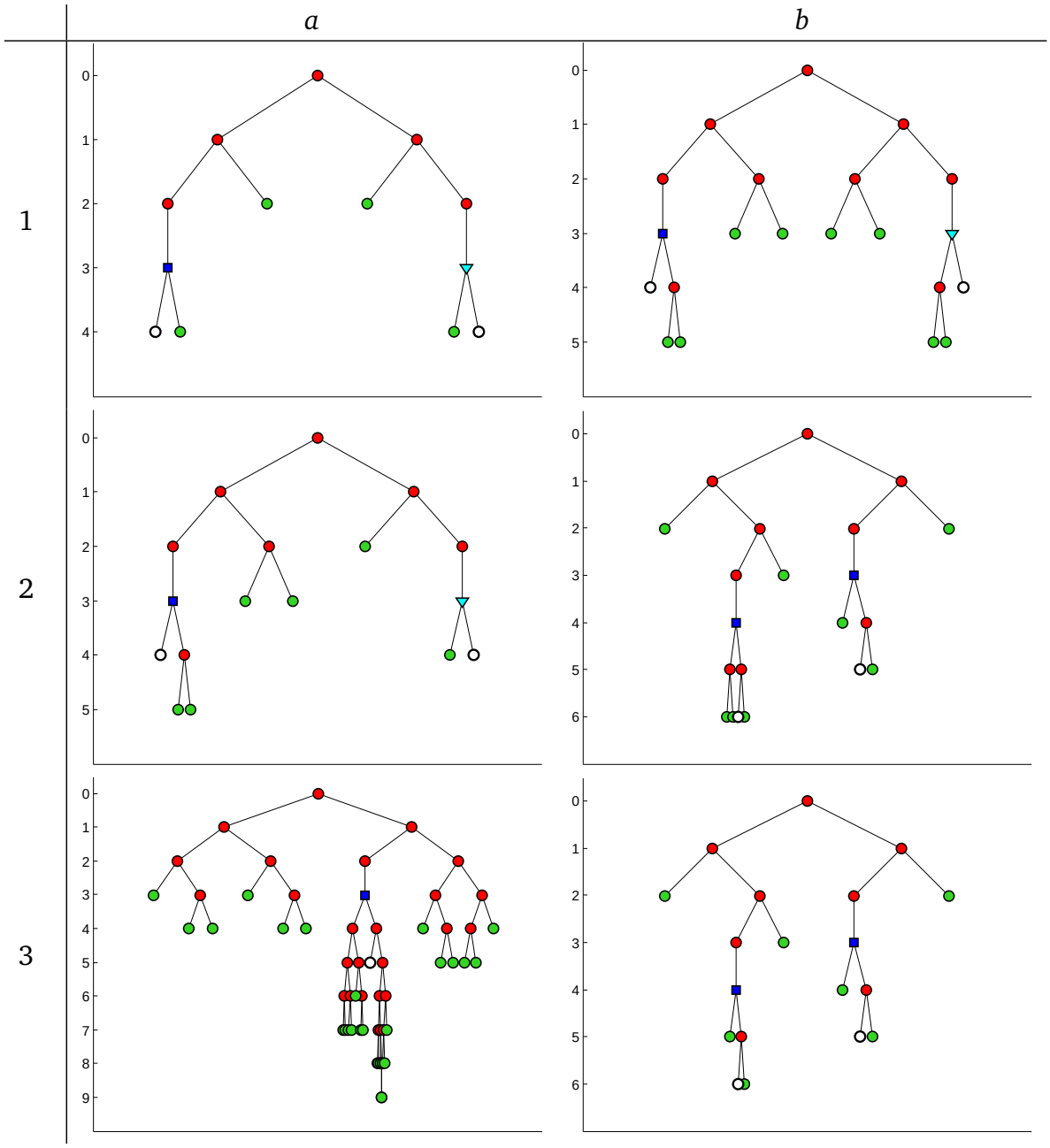


Figure 7.7: $\mathscr{J}$-complete trees for the 3-point scheme and $\mathscr{J} = \{[1], [2]\}$ with different choices of parameters and norms.
The default values are chosen to be norm $\|\cdot\|_2$, LimRadConstr $= 1/100$ and LimRadConstr $= 1/10$ leading to tree 1a.
With norm $\|\cdot\|_\infty$ instead of $\|\cdot\|_2$, tree 1b is computed.
When transforming the basis such that A_1 is diagonal, we get tree 2a instead of 1a.
The trees 2b, 3a and 3b are with respect to the palindromic transform of $\mathscr{A}$. The default values lead to 2b. Norm $\|\cdot\|_w$ with $w = 3$ instead of $\|\cdot\|_2$ leads to tree 3a.
Tree 3b is computed with LimRadConstr $= 1/100$, comparison with 2b shows a little difference below the left negative node.

7.3 Primal 4-Point-Scheme

The parametrized 4-point scheme proposed in [DLG87] is a generalized version of the interpolatory four-point scheme introduced by Dubuc in [Dub86]. This scheme was used as a benchmark problem for the implementation of Algorithm 4.3. [HMR09] determined explicitly the set of parameter values ω for which the scheme is C^1, namely $(0, \omega_*)$ where $\omega_* \approx 0.19273$ is the unique real solution of the equation $32\omega^3 + 4\omega - 1 = 0$. The parametrized family of schemes is given by the symbol

$$a_\omega(z) = -\omega\left(1+z^6\right) + \left(\frac{1}{2}+\omega\right)\left(z^2+z^4\right) + z^3.$$

In order to determine Hölder regularity for values in $(0, \omega_*)$, consider the subdivision matrices of the scheme with symbol $\frac{2}{(z+1)^2}a_\omega(z)$, which are

$$A_1^\omega = \begin{pmatrix} 4\omega & 4\omega & 0 & 0 \\ -2\omega & 1-4\omega & -2\omega & 0 \\ 0 & 4\omega & 4\omega & 0 \\ 0 & -2\omega & 1-4\omega & -2\omega \end{pmatrix}, \; A_2^\omega = \begin{pmatrix} -2\omega & 1-4\omega & -2\omega & 0 \\ 0 & 4\omega & 4\omega & 0 \\ 0 & -2\omega & 1-4\omega & -2\omega \\ 0 & 0 & 4\omega & 4\omega \end{pmatrix}.$$

For $\omega = 1/16$, the scheme corresponds to the Dubuc 4-point scheme. In that case, $\mathscr{A}^\omega = \{A_1^\omega, A_2^\omega\}$ is reducible such that the problem could be reduced to lower dimension. However, the Dubuc 4 point scheme is already known to be C_*^2, see for example [HS08]. For other values of ω, the Hölder regularity is to be determined, and $\mathscr{A}^\omega$ corresponds to the scaled kernel.

Conjecturing $\hat{\rho}(\mathscr{A}^\omega) = \max\left\{\rho(A_1^\omega), \rho\left(A_{[1,2]}^\omega\right)^{\frac{1}{2}}\right\}$, we select 1000 equidistantly sampled parameter values $\omega \subset (0, \omega_*)$ with $\frac{1}{1000} \le \omega \le \omega_* - 10^{-13}$ to validate the conjecture for these cases.

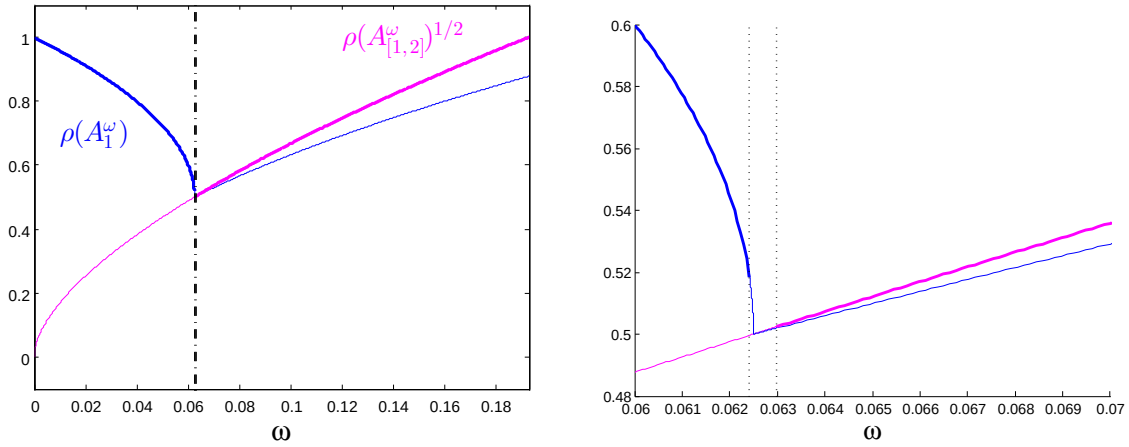


Figure 7.8: Lower bounds for the JSR of the 4-point scheme in dependency of ω which are conjectured to be sharp (*left*), and a zoom on a small neighborhood of $\omega = \frac{1}{16}$ (*right*). The dotted lines indicate the closest lower and higher value of ω for which the conjecture was validated.

Denote by ω_i the i-th of ordered samples, with $\omega_1 = \frac{1}{1000}$ and $\omega_{1000} = \omega_* - 10^{-13}$. Due to

$$\omega_{321} \approx 0.0624 < \frac{1}{16} < \omega_{322} \approx 0.0626,$$

we choose for $\mathscr{A}^{\omega_i}$ the generator set $\mathscr{J} = \{[1],[2]\}$ if $i \leq 321$ and $\mathscr{J} = \{[1,2]\}$ otherwise. See also the visualization of the conjecture in Figure 7.8.

For 999 out of 1000 samples, the conjecture was validated. For $i \leq 321$ and $i \geq 325$, the computation was successful with parameter values LIMRADCOMP = LIMRADCONSTR $= 1/1000$, MAXLEVEL $= 100$, STARTLEVEL $= 3$, NEGLIMIT $= 1$ and $\|\cdot\|_{\text{poly}}$ with IT $= 15$. Figure 7.10 shows $\mathscr{J}$-complete trees descending[2] from [1] for different parameter values ω_i.
For $i \in \{323, 324\}$, the algorithm terminated with output "no result" for this choice of parameters, but successfully found a $\mathscr{J}$-complete when choosing IT $= 25$. For the remaining sample $i = 322$, the algorithm terminated with output "no result" for IT $= 25, 30, 35$.

We thus observe that the computations require more effort for $\omega \searrow \frac{1}{16}$. For $\omega = \frac{1}{16}$, both $[1,2]$ and $[1]$ are strong generators. The spectral radius of the weak generator $[1]$ converges towards 1. The trees displayed in figure 7.10 show that the branch generated by powers of A_1^ω grows. But including $[1]$ in $\mathscr{J}$ in order to subsume these branches was not successful. The runtime rose due to more possibilities in the search tree update but the algorithm led to similar results. This probably results from the fact that the weak generator $A_{[1]}^\omega$ has for $\omega \searrow \frac{1}{16}$ a complex conjugate pair of leading eigenvalues. Possibly, the computation of upper bounds in this case does not lead to sufficiently tight results such that 1-bounded nodes were not detected.
Interestingly, there was only a slightly increasing difficulty to observe for $\omega \nearrow \frac{1}{16}$ although the situation is similar with $[1,2]$ becoming a strong generator. This may result from the ratios of the leading eigenvalues of weak and strong generator: If $\rho(A_1^\omega)$ and $\rho(A_{[1,2]}^\omega)^{\frac{1}{2}}$ are close, the scaled family possesses a weak generator with a spectral radius close to 1 such that, for powers of the weak generator matrix, decay of the norm is slow. Hence, long paths have to be computed. Figure 7.9 shows the spectral radii of the strong and the weak generator matrix in dependance of ω, indicating that the weak generator is very close to being strong for $\omega \searrow \frac{1}{16}$. The runtime increases for $\omega \to \frac{1}{16}$. On a standard PC, it was on a subsecond time-scale for $i \leq 320$ and $i \geq 330$. For $\omega \nearrow \frac{1}{16}$, the runtime was less than 3 minutes. In contrast, the computation for $i = 328$ took 6 minutes and for $i = 325$ 18 minutes.

The length of the vector of matrix balls $\mathbf{A}_{J,r}^\omega$ which is used for computing the upper bound of a norm, compare Chapter 4, indicates how fast the powers $(A_J^\omega)^k$ converge. The computational effort for the computation in each node increases with the length. Interestingly, the difficult parameters slightly bigger than $\frac{1}{16}$ do not possess very long ball vectors. For example, it is of length 9 in case ω_{324},

[2] Since the family is palindromic, we can apply Proposition 6.10.

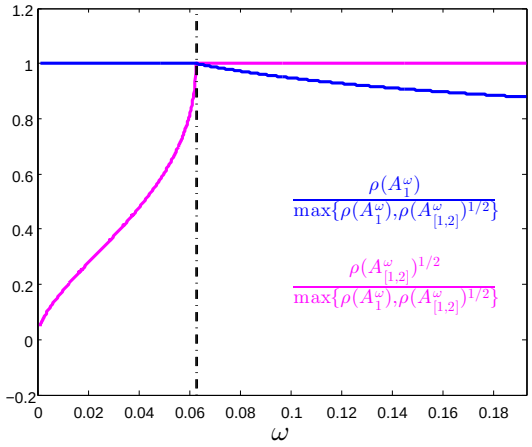


Figure 7.9: Visualization of the spectral radii corresponding to the weak and strong generator matrices of the 4-point scheme, in dependance of ω.

which is short in contrast to length 97 in case of ω_{321} while, using the default parameters, the runtime for ω_{324} was approx. 34 minutes and the runtime for ω_{321} less than 3 minutes.

Comparing the results for the norms $\|\cdot\|_{\text{poly}}$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$ shows that $\|\cdot\|_{\text{poly}}$ is the best choice for the parametrized family. See Figure 7.11 for an exemplary visual comparison. Although there are parameter values ω for which the difference in the trees is negligible, $\|\cdot\|_2$ and $\|\cdot\|_\infty$ were unsuccessful for certain parameter values, as for example ω_{328}. While $\|\cdot\|_{\text{poly}}$ with IT $= 15$ leads in this case to a tree with depth 24, the algorithm terminates for both of the other norms in MAXLEVEL $= 100$ with "no result".

For $\omega > \frac{1}{16}$, the family apparently possesses a spectral gap at 1 since $\bar{J} = [2,1]$ is a cyclic permutation of the strong generator $\bar{J} = [1,2]$. To achieve such a situation for $\omega < \frac{1}{16}$, the palindromic transform of the family is to be considered. Figure 7.12 allows to compare the $\mathcal{J}$-complete[3] trees of ω_{300} with $\|\cdot\|_{\text{poly}}$ before and after transformation as well as results obtained with $\|\cdot\|_w$. It illustrates that the palindromic transform does not necessarily lead to trimmer or shorter trees. But it allows to compute a $\mathcal{J}$-complete tree with only one negative child when using $\|\cdot\|_w$. Comparison for two different values shows that choosing w appropriately does not imply choosing it large.

Figure 7.13 allows to compare trees computed with different STARTLEVEL and norms $\|\cdot\|_{\text{poly}}$ generated with different IT. It additionally shows a tree which was computed with usage of a weak generator (*second row, on the right*).

[3] Proposition 6.10 is not applicable when choosing $\|\cdot\|_w$.

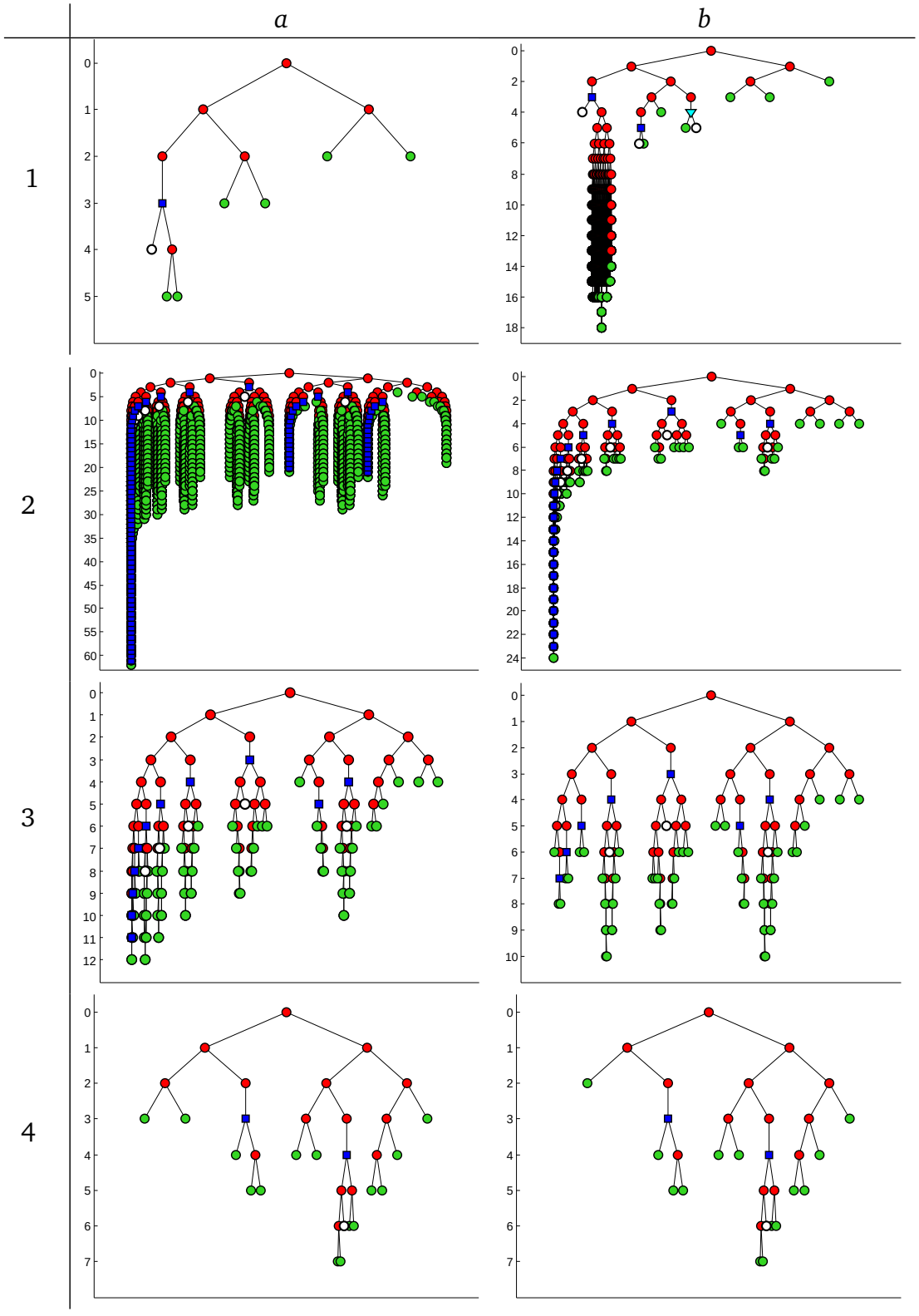


Figure 7.10: $\mathcal{J}$-complete trees descending from $[1]$ of the 4-point scheme obtained for different values ω_i:
With $\mathcal{J} = [1], [2]$: 1a for ω_1, ω_{100} and ω_{300}, 1b for ω_{321}.
With $\mathcal{J} = [1, 2]$: 2a for ω_{325}, 2b for ω_{328}, 3b for ω_{350},
3c for ω_{400}, 4a for ω_{600}, 4b for ω_{1000}.

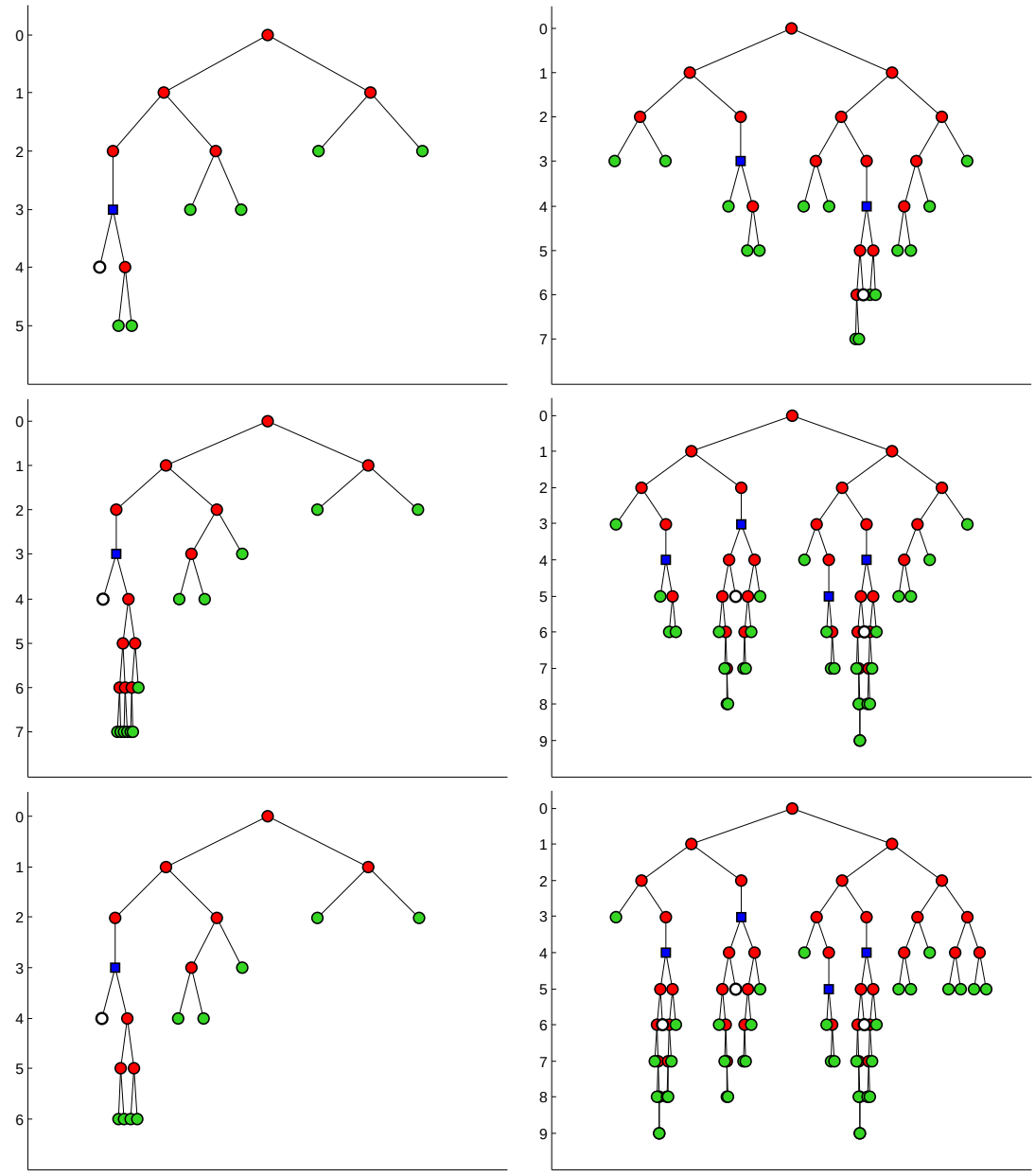


Figure 7.11: $\mathscr{J}$-complete trees descending from $[1]$ of the 4-point scheme for ω_{300} (*left*) and ω_{600} (*right*).
First row: $\|\cdot\|_{\text{poly}}$ with $\text{IT} = 15$
Second row: $\|\cdot\|_2$
Third row: $\|\cdot\|_\infty$

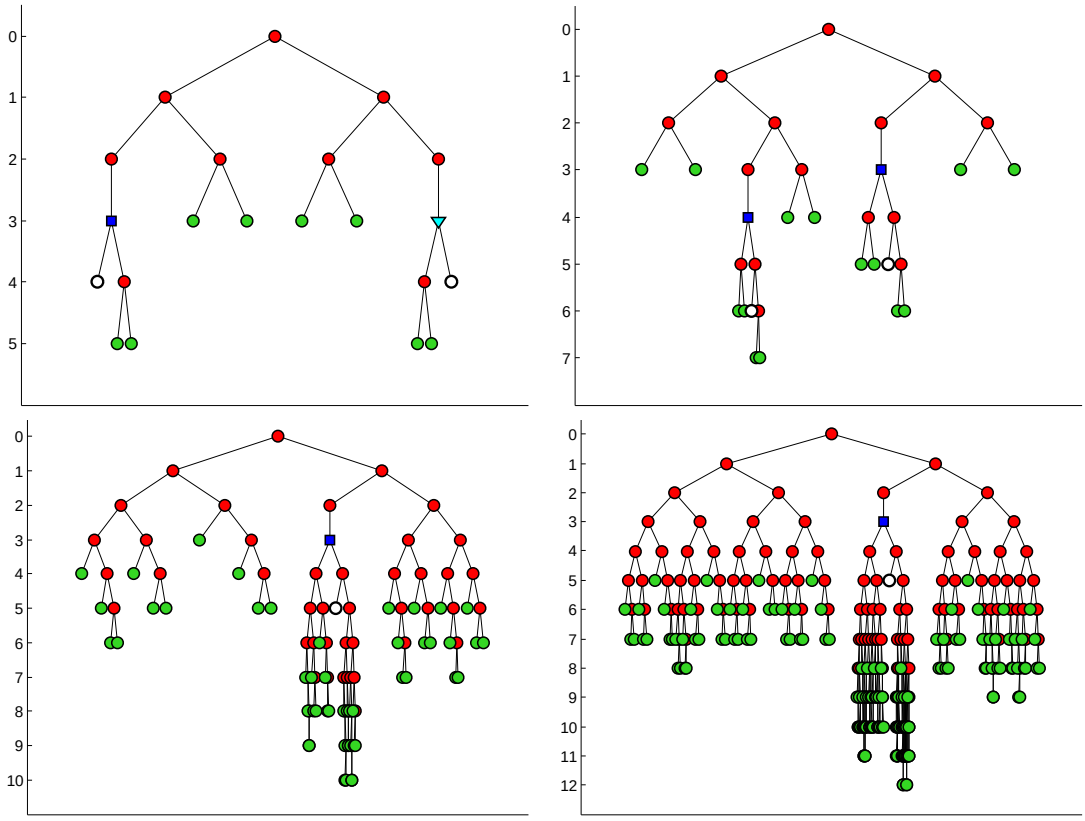


Figure 7.12: $\mathcal{J}$-complete trees of the 4-point scheme resp./ its palindromic transform for ω_{300}.
First row: for $\|\cdot\|_{\text{poly}}$ before (*left*) and after (*right*) palindromic transformation
Second row: after palindromic transformation for $\|\cdot\|_w$, with $w = 3$ (*left*) and $w = 5$ (*right*)

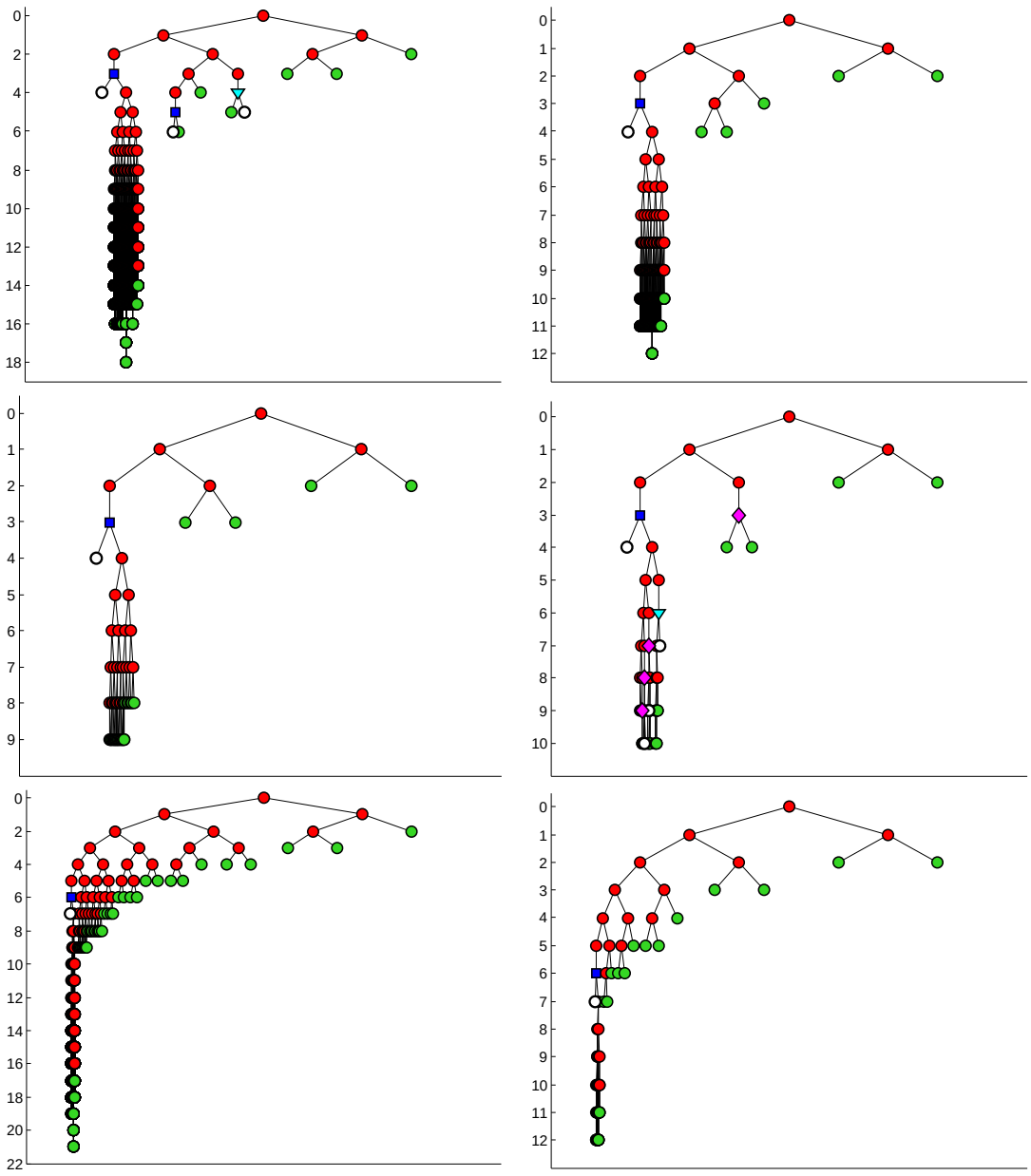


Figure 7.13: $\mathscr{J}$-complete trees descending from $[1]$ of the 4-point scheme with ω_{321} and $\|\cdot\|_{\text{poly}}$ and, unless indicated otherwise, $\mathscr{J} = [1], [2]$ and STARTLEVEL $= 3$.
First row: IT $= 15$ (*left*), IT $= 20$ (*right*)
Second row: IT $= 25$ (*left*), IT $= 25$ and $\mathscr{J} = \{[1], [2], [1, 2]\}$ (*right*)
Third row: IT $= 15$ and STARTLEVEL $= 6$ (*left*), IT $= 25$ and STARTLEVEL $= 6$ (*right*)

7.4 Dual 4-Point-Scheme

[DFH05] proposes a parametrized family of dual 4-point schemes with symbol

$$a_\omega(z) = \frac{(1+z)^3}{4}\left(z^{-1} + 4\omega(-5z + 8 - 6z^{-1} + 8z^{-2} - 5z^{-3})\right).$$

According to [DFH05], the scheme is C^2 for ω in the range of $(0, \frac{1}{48}]$. For $\omega = \frac{1}{128}$, Hölder regularity is $4 - \log_2(9) \approx 2.8301$, as shown by [HS08]. The matrix family $\mathscr{A}^\omega = \{A_1^\omega, A_2^\omega\}$ deduced from the scaled kernel is given by

$$A_1^\omega = \begin{pmatrix} 32\omega & 32\omega & 0 & 0 \\ -20\omega & 1-24\omega & -20\omega & 0 \\ 0 & 32\omega & 32\omega & 0 \\ 0 & -20\omega & 1-24\omega & -20\omega \end{pmatrix}, \quad A_2^\omega = \begin{pmatrix} -20\omega & 1-24\omega & -20\omega & 0 \\ 0 & 32\omega & 32\omega & 0 \\ 0 & -20\omega & 1-24\omega & -20\omega \\ 0 & 0 & 32\omega & 32\omega \end{pmatrix}.$$

We sample $\omega \subset (0, \frac{1}{48}]$ equidistantly by 100 parameter values ω_i such that

$$\frac{1}{1000} \leq \omega_i \leq \frac{1}{48}.$$

Denote by ω_i the i-th of ordered samples. The conjecture

$$\hat{\rho}(\mathscr{A}^\omega) = \max\left\{\rho(A_1^\omega), \rho\left(A_{[1,2]}^\omega\right)^{\frac{1}{2}}\right\}$$

is visualized in Figure 7.14, and was validated for $i \leq 35$ as well as for $i \geq 66$. In between, the computation was not successful. For these parameter values, the leading eigenvalue of the generator matrix is not real. The exemplary $\mathscr{J}$-complete trees descending from [1] for ω_{34} and ω_{68} in Figure 7.15 were computed with upper bounds of norms which are determined via bounds of the eigenvalues instead of matrix balls, see Section 4.3.2.

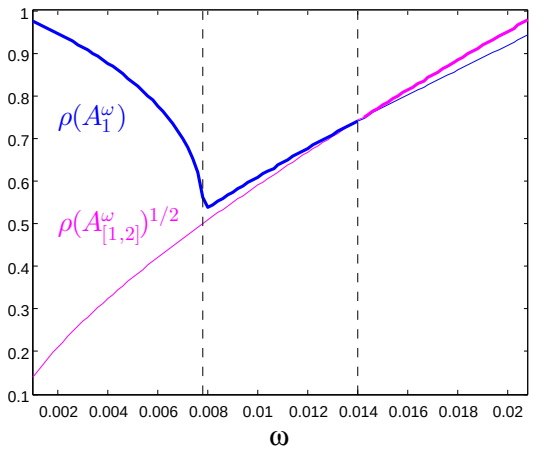


Figure 7.14: Lower bounds for the JSR of the dual 4-point scheme in dependency of ω which are conjectured to be sharp (*left*). The dotted lines indicate the closest lower and higher value of ω for which the conjecture was validated.

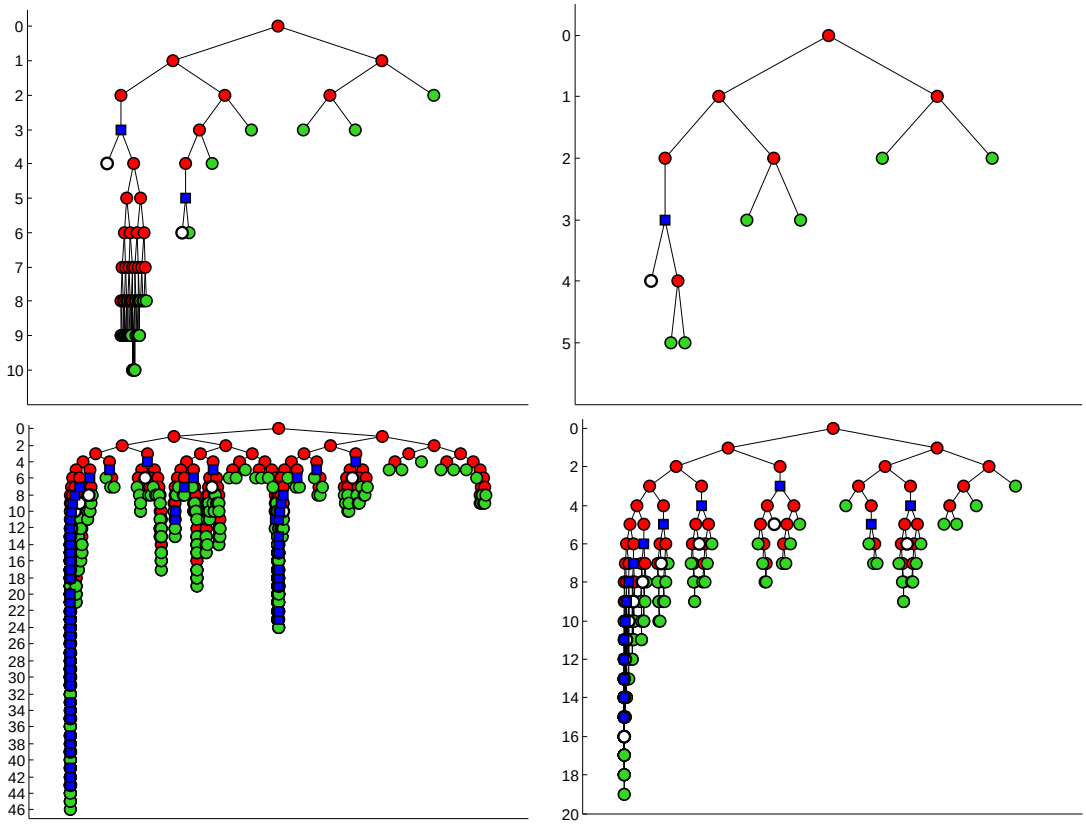


Figure 7.15: $\mathcal{J}$-complete trees descending from $[1]$ of the dual 4-point scheme.
First row: ω_{34} with $\mathcal{J} = \{[1],[2]\}$ and $\|\cdot\|_2$ (*left*)
resp. $\|\cdot\|_{\text{poly}}$ with $\text{IT} = 10$ (*right*)
Second row: ω_{68} with $\mathcal{J} = \{[1,2]\}$ and $\|\cdot\|_2$ (*left*)
resp. $\|\cdot\|_{\text{poly}}$ with $\text{IT} = 10$ (*right*)

7.5 DD 6-point scheme

The Dubuc-Deslaurier (DD) 6-point scheme proposed in [DD89] is a symmetric interpolatory subdivision scheme which reproduces polynomials of degree 5. It has the symbol

$$a(z) = \frac{1}{256}\left(3\left(z^{-5}+z^{5}\right) - 25\left(z^{-3}+z^{3}\right) + 150\left(z^{-1}+z\right) + 1\right),$$

and the scaled kernel $b(z) = \frac{2^5}{(1+z)^6}a(z)$ leads to $\mathcal{A} = \{A_1, A_2\}$ with

$$A_1 = \frac{1}{8}\begin{pmatrix} -18 & -18 & 0 & 0 \\ 3 & 38 & 3 & 0 \\ 0 & -18 & -18 & 0 \\ 0 & 3 & 38 & 3 \end{pmatrix}, \quad A_2 = \frac{1}{8}\begin{pmatrix} 3 & 38 & 3 & 0 \\ 0 & -18 & -18 & 0 \\ 0 & 3 & 38 & 3 \\ 0 & 0 & -18 & -18 \end{pmatrix}.$$

With $\mathcal{J} = \{[1],[2]\}$, a $\mathcal{J}$-complete tree with $\|\cdot\|_2$ is found and displayed in Figure 7.16 *top left*. Palindromic transformation leads to the tree *top right* for

$\|\cdot\|_2$, and to trees with only one negative child for $\|\cdot\|_w$ with parameter $w = 3$ *bottom left* and for $\|\cdot\|_w$ with parameter $w = 10$ *bottom right*. Comparison of these two figures shows that choosing w large is not always to recommend.

We deduce that $\hat{\rho}(\mathscr{A}) = \rho(A_1) = \frac{9}{2}$. Therewith, the DD 6-point scheme has Hölder regularity $\beta = 5 - \log_2(\hat{\rho}(\mathscr{A})) \approx 2.8301$.

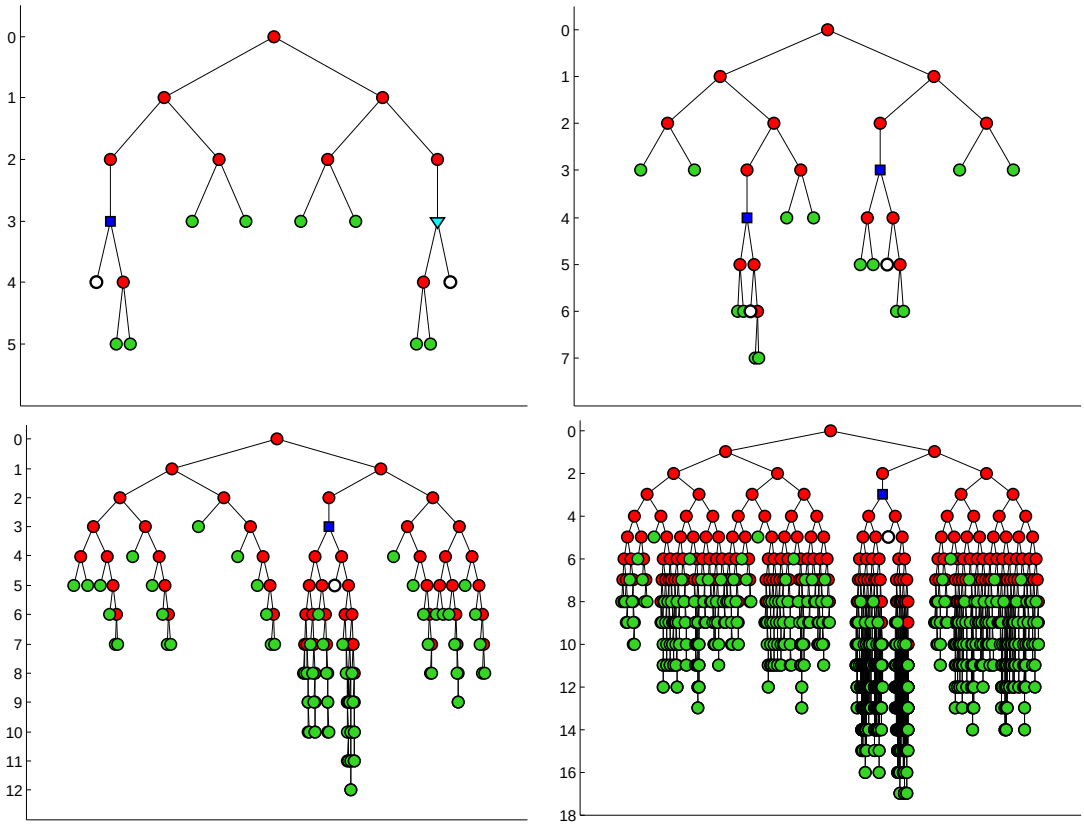


Figure 7.16: $\mathscr{J}$-complete trees of the DD 6-point scheme with $\mathscr{J} = \{[1], [2]\}$
First row: $\|\cdot\|_2$ before (*left*) and after palindromic transform (*right*)
Second row: $\|\cdot\|_w$ after palindromic transform with $w = 3$ (*left*) resp. $w = 10$ (*right*)

7.6 DD 8-point scheme

The 8-point scheme proposed by Deslaurier and Dubuc ([DD89]) has the symbol

$$a(z) = \frac{\left(-5\left(z^{-7}+z^{7}\right)+49\left(z^{-5}+z^{5}\right)-245\left(z^{-3}+z^{3}\right)+1225\left(z^{-1}+z\right)+1\right)}{2048}.$$

Considering $\frac{2^5}{(1+z)^6}a(z)$, we obtain the subdivision matrices $\mathscr{A} = \{A_1, A_2\}$ with

$$A_1 = \frac{1}{64}\begin{pmatrix} 30 & -14 & -14 & 30 & 0 & 0 & 0 & 0 \\ -5 & -56 & 154 & -56 & -5 & 0 & 0 & 0 \\ 0 & 30 & -14 & -14 & 30 & 0 & 0 & 0 \\ 0 & -5 & -56 & 154 & -56 & -5 & 0 & 0 \\ 0 & 0 & 30 & -14 & -14 & 30 & 0 & 0 \\ 0 & 0 & -5 & -56 & 154 & -56 & -5 & 0 \\ 0 & 0 & 0 & 30 & -14 & -14 & 30 & 0 \\ 0 & 0 & 0 & -5 & -56 & 154 & -56 & -5 \end{pmatrix}$$

$$A_2 = \frac{1}{64}\begin{pmatrix} -5 & -56 & 154 & -56 & -5 & 0 & 0 & 0 \\ 0 & 30 & -14 & -14 & 30 & 0 & 0 & 0 \\ 0 & -5 & -56 & 154 & -56 & -5 & 0 & 0 \\ 0 & 0 & 30 & -14 & -14 & 30 & 0 & 0 \\ 0 & 0 & -5 & -56 & 154 & -56 & -5 & 0 \\ 0 & 0 & 0 & 30 & -14 & -14 & 30 & 0 \\ 0 & 0 & 0 & -5 & -56 & 154 & -56 & -5 \\ 0 & 0 & 0 & 0 & 30 & -14 & -14 & 30 \end{pmatrix}$$

With $\mathscr{J} = \{[1],[2]\}$ and norm $\|\cdot\|_2$, a $\mathscr{J}$-complete tree is found. The results for different parameter values STARTLEVEL and NEGLIMIT are visualized in Figure 7.17. Hence, $\hat{\rho}(\mathscr{A}) = \rho(A_1) \approx 2.7299$. Hölder regularity of the DD 8-point scheme is given by $5 - \log_2(\hat{\rho}(\mathscr{A})) \approx 3.5511$.

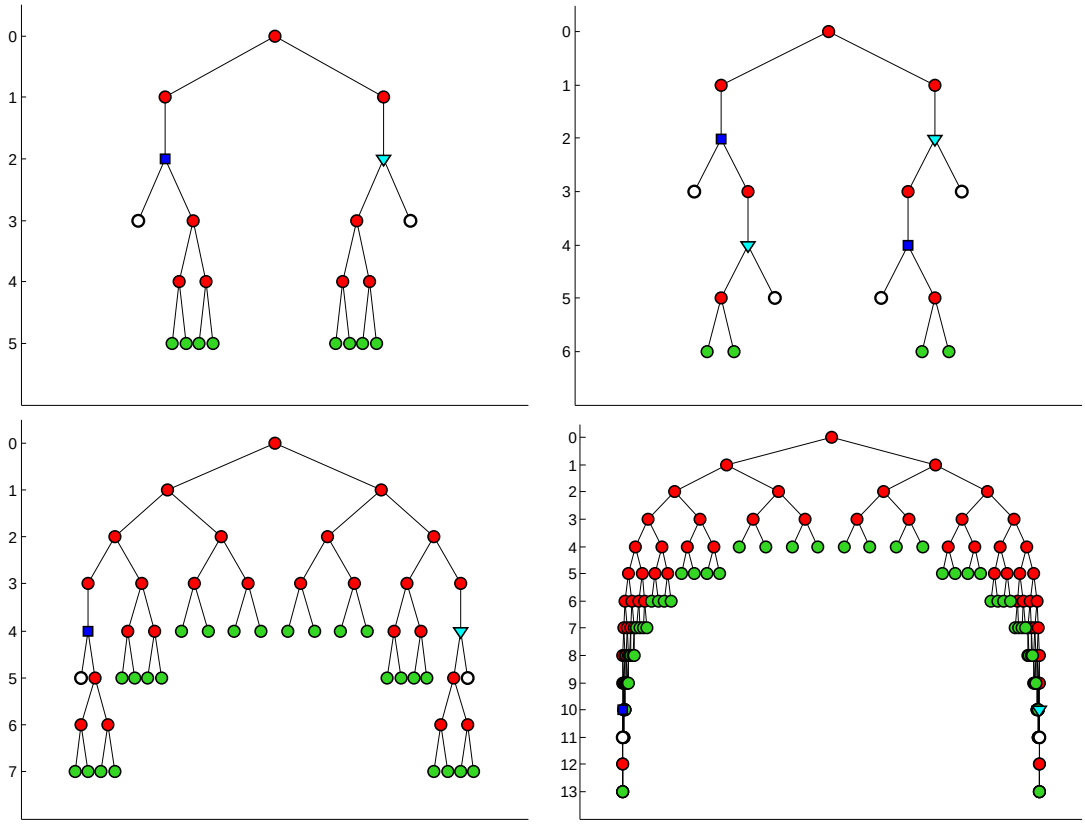


Figure 7.17: $\mathcal{J}$-complete trees for $\mathcal{E}$ with generators $\mathcal{J} = \{[1], [2]\}$ and $\|\cdot\|_2$.
First row: STARTLEVEL $= 2$ and NEGLIMIT $= 1$ (*left*) resp. NEGLIMIT $= 2$ (*right*)
Second row: NEGLIMIT $= 1$ and STARTLEVEL $= 4$ (*left*) resp. STARTLEVEL $=$ 10 (*right*)

7.7 Ternary 4-point scheme

In [HIDS02], a C^2 ternary subdivision scheme with parameter μ is proposed. Its Hölder regularity was analyzed in [ZZYZ07] using Rioul's method [Rio92]. According to [ZZYZ07], the highest smoothness is obtained for $\mu = \frac{1}{11}$ with $2 - \log_3(\frac{9}{11}) \approx 2.1827$. Consider the difference scheme of the second divided differences with mask $\frac{1}{11} \cdot [-4, 5, 9, 5, -4]$. After reducing dimension in order to obtain an irreducible family, this leads to analyzing $\mathscr{A} = \{A_1, A_2, A_3\}$ with

$$A_1 = \frac{1}{11}\begin{pmatrix} 5 & -4 & 0 \\ -4 & 5 & 0 \\ 0 & 9 & 0 \end{pmatrix}, \quad A_2 = \frac{1}{11}\begin{pmatrix} -4 & 5 & 0 \\ 0 & 9 & 0 \\ 0 & 5 & -4 \end{pmatrix}, \quad A_3 = \frac{1}{11}\begin{pmatrix} 0 & 9 & 0 \\ 0 & 5 & -4 \\ 0 & -4 & 5 \end{pmatrix}.$$

For $\mathscr{J} = \{[1], [2], [3]\}$, there is a $\mathscr{J}$-complete tree with respect to norm $\|\cdot\|_2$, see Figure 7.18. We conclude that $\hat{\rho}(\mathscr{A}) = \rho(A_1) = \frac{9}{11}$. Hölder regularity is given by $2 - \log_3(\hat{\rho}(\mathscr{A}))$, confirming the result of [ZZYZ07].

7.8 Quaternary 3-point scheme

A quaternary scheme can be constructed by sampling 4 new points from a quadratic polynomial q which interpolates 3 old points. Choosing the points $q(-\frac{3}{8})$, $q(-\frac{1}{8})$, $q(\frac{1}{8})$ and $q(\frac{3}{8})$ leads to a scheme $S_{\mathbf{a}}$ with mask

$$\mathbf{a} = \frac{1}{128}[-15, -7, 9, 33, 110, 126, 126, 110, 33, 9, -7, -15]$$

with according symbol $a(z)$. From $S_{\mathbf{b}}$ with

$$b(z) = \frac{4}{(1+z+z^2+z^3)^2} \cdot a(z) = \frac{1}{32}\left(-15(z^5+1) + 23(z^4+z) + 8(z^3+z^2))\right),$$

we deduce four (5×5)-matrices, but they share two zero-columns such that they are reducible. Splitting into lower dimensional families leads to only one family $\mathscr{A} = \{A_1, A_2, A_3, A_4\}$ being non-zero, with

$$A_1 = \frac{1}{32}\begin{pmatrix} 23 & -15 & 0 \\ -15 & 23 & 0 \\ 0 & 8 & 0 \end{pmatrix}, \quad A_2 = \frac{1}{32}\begin{pmatrix} -15 & 23 & 0 \\ 0 & 8 & 0 \\ 0 & 8 & 0 \end{pmatrix}$$

$$A_3 = \frac{1}{32}\begin{pmatrix} 0 & 8 & 0 \\ 0 & 8 & 0 \\ 0 & 23 & -15 \end{pmatrix}, \quad A_4 = \frac{1}{32}\begin{pmatrix} 0 & 8 & 0 \\ 0 & 23 & -15 \\ 0 & -15 & 23 \end{pmatrix}$$

With $\mathscr{J} = \{[1], [4]\}$ and $\|\cdot\|_2$, a $\mathscr{J}$-complete tree exists, see Figure 7.19. Hence, $\hat{\rho}(\mathscr{A}) = \rho(A_1) = \frac{19}{16}$ and $S_{\mathbf{a}}$ has Hölder regularity $\beta_* = 1 - \log_4(\hat{\rho}(\mathscr{A})) = 3 - \log_4(19) \approx 0.8760$.

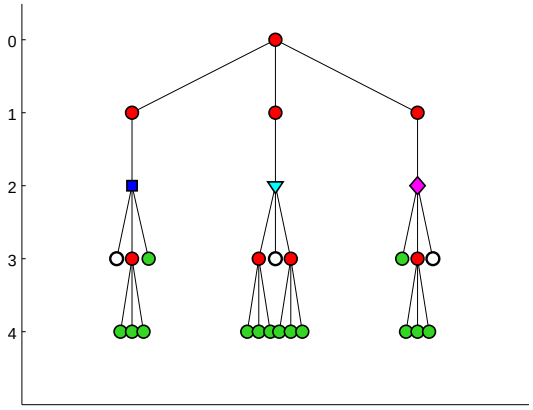


Figure 7.18: $\mathcal{J}$-complete tree of the ternary 4-point scheme with $\mathcal{J} = \{[1], [2], [3]\}$ and norm $\|\cdot\|_2$.

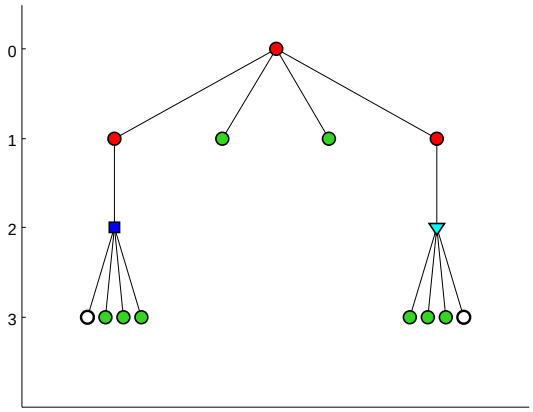


Figure 7.19: Tree of the quaternary 3-point scheme with generators $\mathcal{J} = \{[1], [4]\}$ and norm $\|\cdot\|_2$.

7.9 Lane-Riesenfeld C-schemes

In [CHR13], the Lane-Riesenfeld algorithm of [LR80] is generalized. The so-called C-schemes given by the symbol

$$a(z) = \left(\frac{1+z}{2}\right)^{k+4} \left(\frac{-z^{-1}+10-z}{8}\right)^{k} \left(-z^{-1}+4-z\right)$$

are proposed and lower and upper bounds for their Hölder regularity are derived for $0 \leq k \leq 4$. The set-valued tree approach reveals that the upper bounds are sharp. The scaled kernel is given in dependence of the parameter k by

$$b(z) = \frac{1}{2}\left(\frac{-z^{-1}+10-z}{4}\right)^{k} \left(-z^{-1}+4-z\right).$$

For $0 \leq k \leq 4$, $\mathcal{J}$-complete trees for the subdivision matrices were found when choosing $\mathcal{J} = \{[1],[2]\}$ and $\|\cdot\|_2$. Although the dimension of the matrices increases with k, all examples were computed on a standard PC on a sub-second timescale. For example, the algorithm terminated for $k = 4$ with a pair of (10×10)-matrices in less than 0.3 sec. Comparison with the runtime for the (4×4)-matrices of the 4-point scheme for values close to $\frac{1}{16}$ shows that long runtimes do not necessarily correspond to high-dimensional problems.

The subdivision matrices of the kernel for $k = 0$ are

$$A_1 = \frac{1}{2}\begin{bmatrix} 4 & 0 \\ -1 & -1 \end{bmatrix}, \quad A_2 = \frac{1}{2}\begin{bmatrix} -1 & -1 \\ 0 & 4 \end{bmatrix}.$$

This example can be handled by the 3-member-inequality:

$$\rho(A_1) = 2 \leq \hat{\rho}(\mathcal{A}) \leq 2 = \|A_1\|_\infty = \|A_2\|_\infty$$

The corresponding $\mathcal{J}$-complete tree of the scaled family consists of the nodes $\{\emptyset,[1],[2]\}$ and the scheme has Hölder regularity $3-\log_2(2) = 2$, i.e., the scheme is C^2_*.

For $k = 1$, we obtain the (4×4)-matrices

$$A_1 = \frac{1}{16}\begin{pmatrix} 1 & 42 & 1 & 0 \\ 0 & -14 & -14 & 0 \\ 0 & 1 & 42 & 1 \\ 0 & 0 & -14 & -14 \end{pmatrix}, \quad A_2 = \frac{1}{16}\begin{pmatrix} -14 & -14 & 0 & 0 \\ 1 & 42 & 1 & 0 \\ 0 & -14 & -14 & 0 \\ 0 & 1 & 42 & 1 \end{pmatrix}$$

The corresponding $\mathcal{J}$-complete tree is displayed in Figure 7.20 *top left*. Hence, $\hat{\rho}(\mathcal{A}) = \rho(A_1) \approx 2.5935$, and the scheme has Hölder regularity $4-\log_2(\rho(A_1)) \approx 2.6251$.

For $k = 2$ we obtain the (6×6)-matrices

$$A_1 = \frac{1}{128}\begin{pmatrix} 24 & 448 & 24 & 0 & 0 & 0 \\ -1 & -183 & -183 & -1 & 0 & 0 \\ 0 & 24 & 448 & 24 & 0 & 0 \\ 0 & -1 & -183 & -183 & -1 & 0 \\ 0 & 0 & 24 & 448 & 24 & 0 \\ 0 & 0 & -1 & -183 & -183 & -1 \end{pmatrix}, \quad A_2 = \frac{1}{128}\begin{pmatrix} -1 & -183 & -183 & -1 & 0 & 0 \\ 0 & 24 & 448 & 24 & 0 & 0 \\ 0 & -1 & -183 & -183 & -1 & 0 \\ 0 & 0 & 24 & 448 & 24 & 0 \\ 0 & 0 & -1 & -183 & -183 & -1 \\ 0 & 0 & 0 & 24 & 448 & 24 \end{pmatrix}.$$

The corresponding $\mathscr{J}$-complete tree is displayed in Figure 7.20 *top right*. Hence, $\hat{\rho}(\mathscr{A}) = \rho(A_1) \approx 3.3891$, and the scheme has Hölder regularity $5 - \log_2(\rho(A_1)) \approx 3.2391$.

For $k = 3$ we obtain the (8×8)-matrices

$$A_1 = \frac{1}{1024}\begin{pmatrix} -34 & -2302 & -2302 & -34 & 0 & 0 & 0 & 0 \\ 1 & 424 & 4846 & 424 & 1 & 0 & 0 & 0 \\ 0 & -34 & -2302 & -2302 & -34 & 0 & 0 & 0 \\ 0 & 1 & 424 & 4846 & 424 & 1 & 0 & 0 \\ 0 & 0 & -34 & -2302 & -2302 & -34 & 0 & 0 \\ 0 & 0 & 1 & 424 & 4846 & 424 & 1 & 0 \\ 0 & 0 & 0 & -34 & -2302 & -2302 & -34 & 0 \\ 0 & 0 & 0 & 1 & 424 & 4846 & 424 & 1 \end{pmatrix}$$

$$A_2 = \frac{1}{1024}\begin{pmatrix} 1 & 424 & 4846 & 424 & 1 & 0 & 0 & 0 \\ 0 & -34 & -2302 & -2302 & -34 & 0 & 0 & 0 \\ 0 & 1 & 424 & 4846 & 424 & 1 & 0 & 0 \\ 0 & 0 & -34 & -2302 & -2302 & -34 & 0 & 0 \\ 0 & 0 & 1 & 424 & 4846 & 424 & 1 & 0 \\ 0 & 0 & 0 & -34 & -2302 & -2302 & -34 & 0 \\ 0 & 0 & 0 & 1 & 424 & 4846 & 424 & 1 \\ 0 & 0 & 0 & 0 & -34 & -2302 & -2302 & -34 \end{pmatrix}.$$

The corresponding $\mathscr{J}$-complete tree is displayed in Figure 7.20 *bottom left*. Hence, $\hat{\rho}(\mathscr{A}) = \rho(A_1) \approx 4.4567$, and the scheme has Hölder regularity $6 - \log_2(\rho(A_1)) \approx 3.844$.

For $k = 4$ we obtain the (10×10)-matrices

$$A_1 = \frac{1}{8192}\begin{pmatrix} 44 & 6576 & 53064 & 6576 & 44 & 0 & 0 & 0 & 0 & 0 \\ -1 & -765 & -28290 & -28290 & -765 & -1 & 0 & 0 & 0 & 0 \\ 0 & 44 & 6576 & 53064 & 6576 & 44 & 0 & 0 & 0 & 0 \\ 0 & -1 & -765 & -28290 & -28290 & -765 & -1 & 0 & 0 & 0 \\ 0 & 0 & 44 & 6576 & 53064 & 6576 & 44 & 0 & 0 & 0 \\ 0 & 0 & -1 & -765 & -28290 & -28290 & -765 & -1 & 0 & 0 \\ 0 & 0 & 0 & 44 & 6576 & 53064 & 6576 & 44 & 0 & 0 \\ 0 & 0 & 0-1 & -765 & -28290 & -28290 & -765 & -1 & 0 & \\ 0 & 0 & 0 & 0 & 44 & 6576 & 53064 & 6576 & 44 & 0 \\ 0 & 0 & 0 & 0 & -1 & -765 & -28290 & -28290 & -765 & -1 \end{pmatrix}$$

$$A_2 = \frac{1}{8192}\begin{pmatrix} -1 & -765 & -28290 & -28290 & -765 & -1 & 0 & 0 & 0 & 0 \\ 0 & 44 & 6576 & 53064 & 6576 & 44 & 0 & 0 & 0 & 0 \\ 0 & -1 & -765 & -28290 & -28290 & -765 & -1 & 0 & 0 & 0 \\ 0 & 0 & 44 & 6576 & 53064 & 6576 & 44 & 0 & 0 & 0 \\ 0 & 0 & -1 & -765 & -28290 & -28290 & -765 & -1 & 0 & 0 \\ 0 & 0 & 0 & 44 & 6576 & 53064 & 6576 & 44 & 0 & 0 \\ 0 & 0 & 0-1 & -765 & -28290 & -28290 & -765 & -1 & 0 & \\ 0 & 0 & 0 & 0 & 44 & 6576 & 53064 & 6576 & 44 & 0 \\ 0 & 0 & 0 & 0 & -1 & -765 & -28290 & -28290 & -765 & -1 \\ 0 & 0 & 0 & 0 & 0 & 44 & 6576 & 53064 & 6576 & 44 \end{pmatrix}.$$

The corresponding $\mathscr{J}$-complete tree is displayed in Figure 7.20 *bottom right*. Hence, $\hat{\rho}(\mathscr{A}) = \rho(A_1) \approx 5.8917$, and the scheme has Hölder regularity $7 - \log_2(\rho(A_1)) \approx 4.4413$.

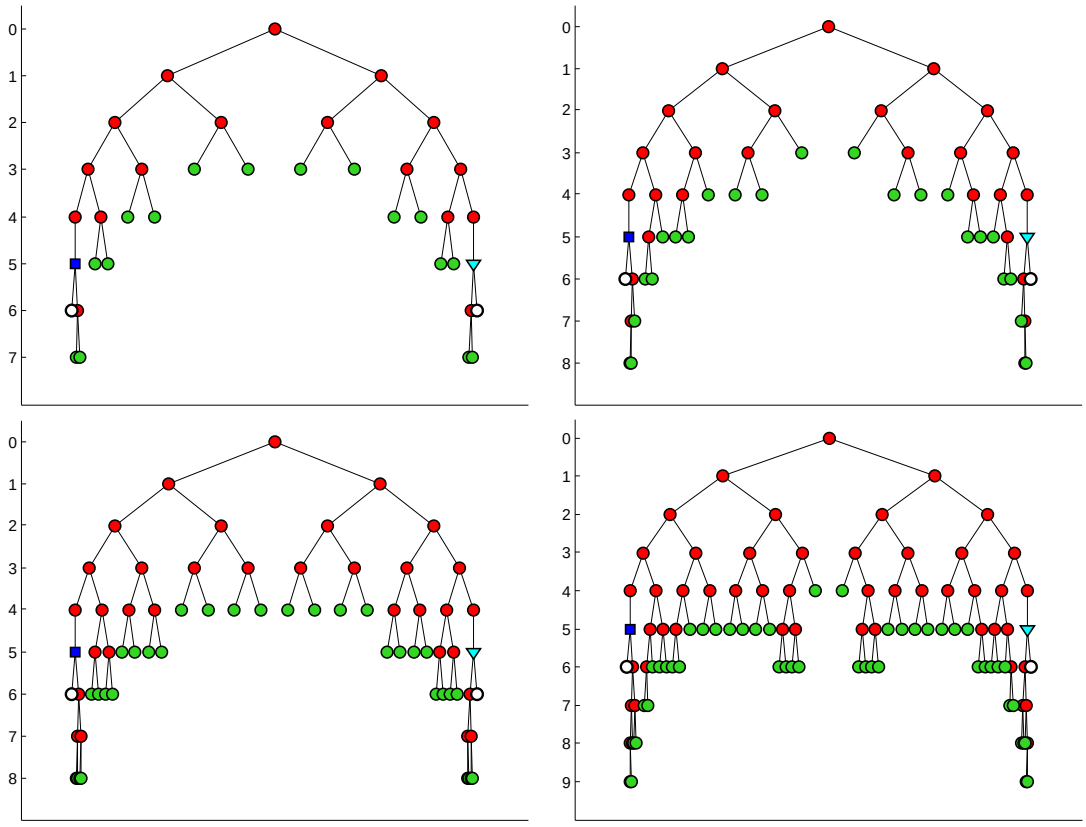


Figure 7.20: $\mathscr{J}$-complete trees for the Lane-Riesenfeld C-schemes and $\mathscr{J} = \{[1],[2]\}$, computed with $\|\cdot\|_2$.
First row: (*left*) $k = 1$, (*right*) $k = 2$
Second row: (*left*) $k = 3$, (*right*) $k = 4$

7.10 A parametrized 8-point scheme

The parametrized 8-point scheme with symbol

$$a_\omega(z) = -\omega\cdot\left(z^{-7}+z^7\right)+(5\omega+\frac{3}{256})\cdot\left(z^{-5}+z^5\right)-(9\omega+\frac{25}{256})\cdot\left(z^{-3}+z^3\right)$$
$$+(5\omega+\frac{75}{128})\cdot\left(z^{-1}+z\right)+1$$

is a linear blend of the DD 6-point scheme, obtained for $\omega = 0$, and the DD 8-point scheme, obtained for $\omega = \frac{5}{2048}$. Although these schemes are only C^2 respective C^3, there are values ω such that the parametrized 8-point scheme is C^4, as we could numerically validate by means of Algorithm 4.12. To this aim, the scheme with

symbol $\frac{2^4}{(z+1)^5}a(z)$ is to be checked for contractivity. The subdivision matrices to analyze are given by

$$A_1^\omega = \begin{pmatrix}
192w & 832w-9/4 & 832w-9/4 & 192w & 0 & 0 & 0 & 0 \\
-32w & -512w+3/8 & -960w+\frac{19}{4} & -512w+3/8 & -32w & 0 & 0 & 0 \\
0 & 192w & 832w-9/4 & 832w-9/4 & 192w & 0 & 0 & 0 \\
0 & -32w & -512w+3/8 & -960w+\frac{19}{4} & -512w+3/8 & -32w & 0 & 0 \\
0 & 0 & 192w & 832w-9/4 & 832w-9/4 & 192w & 0 & 0 \\
0 & 0 & -32w & -512w+3/8 & -960w+\frac{19}{4} & -512w+3/8 & -32w & 0 \\
0 & 0 & 0 & 192w & 832w-9/4 & 832w-9/4 & 192w & 0 \\
0 & 0 & 0 & -32w & -512w+3/8 & -960w+\frac{19}{4} & -512w+3/8 & -32w
\end{pmatrix}$$

$$A_2^\omega = \begin{pmatrix}
-32w & -512w+3/8 & -960w+\frac{19}{4} & -512w+3/8 & -32w & 0 & 0 & 0 \\
0 & 192w & 832w-9/4 & 832w-9/4 & 192w & 0 & 0 & 0 \\
0 & -32w & -512w+3/8 & -960w+\frac{19}{4} & -512w+3/8 & -32w & 0 & 0 \\
0 & 0 & 192w & 832w-9/4 & 832w-9/4 & 192w & 0 & 0 \\
0 & 0 & -32w & -512w+3/8 & -960w+\frac{19}{4} & -512w+3/8 & -32w & 0 \\
0 & 0 & 0 & 192w & 832w-9/4 & 832w-9/4 & 192w & 0 \\
0 & 0 & 0 & -32w & -512w+3/8 & -960w+\frac{19}{4} & -512w+3/8 & -32w \\
0 & 0 & 0 & 0 & 192w & 832w-9/4 & 832w-9/4 & 192w
\end{pmatrix}$$

The lower bounds of the tree member inequality allow to localize a promising parameter interval, see Figure 7.22. For 8 values $\omega \in (0.003341, 0.003989)$, a contractive tree was found with $\|\cdot\|_2$. One of these trees is exemplarily shown in Figure 7.21.

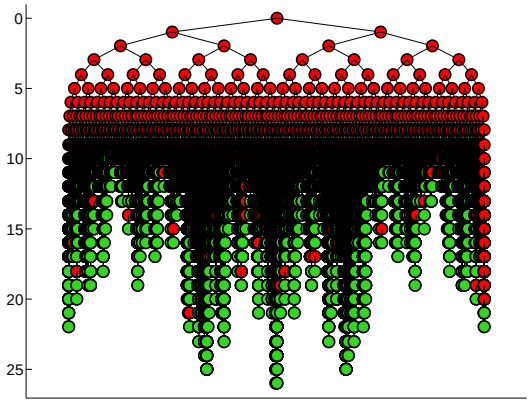


Figure 7.21: Contractive tree for the 8-point scheme with $\omega = 0.00375$ and $\|\cdot\|_2$.

The runtime of the computation on a standard PC varied for the 8 samples between 3 seconds for $\omega \approx 0.003797$ leading to a contractive tree with 33362 nodes and 51 hours for $\omega \approx 0.0033411$ and a tree with $2 \cdot 10^9$ nodes.

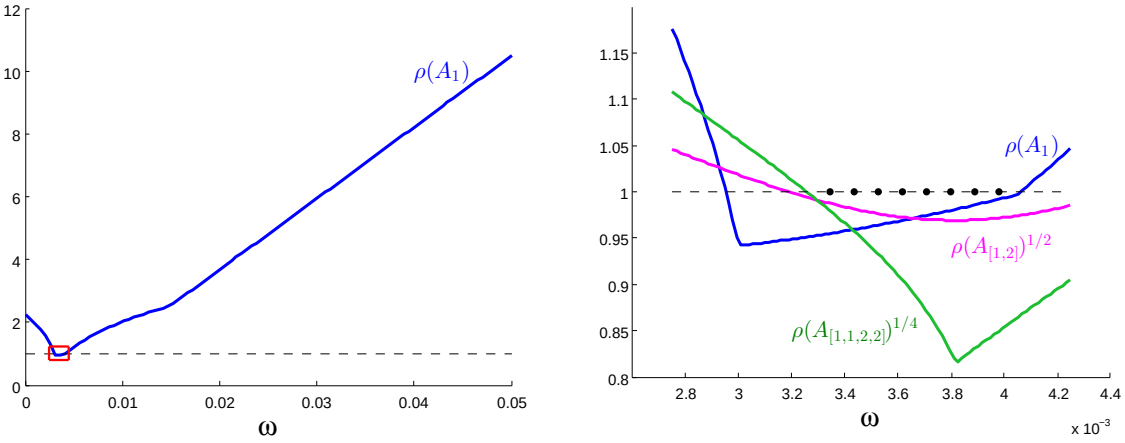


Figure 7.22: Lower bound for the JSR of the unscaled 8-point scheme family in dependency of ω (*left*) and a zoom on the promising interval in the red box with additional lower bounds (*right*).
The black dots indicate parameter values for which we validated contractivity numerically.

We conjecture that the lower bounds for the JSR as visualized in Figure 7.22 (*right*) are sharp. Nevertheless, there is no confirmation by Algorithm 4.3 despite various trials to compute a $\mathscr{J}$-complete tree with different choices of norms and parameters. The blended schemes DD 6-point and DD 8-point itself are not difficult to compute.

Under the premise that our conjecture for the FP-products is true, Figure 7.22 indicates for which values of ω the smoothness is maximized. For the promising value $\omega_* \approx 0.0036563$, we could improve the lower bound for the Hölder exponent: When applying Algorithm 4.12 to the family $\frac{1}{0.98} \cdot \mathscr{A}^{\omega_*}$, a contractive tree was found. Therewith, $\hat{\rho}(\mathscr{A}^{\omega_*}) < 0.98$ such that a lower bound for Hölder regularity of $S_{\mathbf{a}_{\omega_*}}$ is given by $4 - \log_2(0.98) \approx 4.0291$.

8 Conclusion

The proposed strategy for joint spectral radius determination is an alternative to computing an extremal norm. Both approaches can only be successful if the considered matrix family has the finiteness property since they attempt to validate that a conjectured product is FP-product. The set-valued tree method offers a sufficient criterion, namely the existence of a $\mathscr{J}$-complete tree, which is satisfiable also in case of more than one FP-product. Additionally, these products do not have to meet requirements as asymptotic simplicity as demonstrated by settling the examples proposed in [GWZ05]. In theory, even product boundedness or irreducibility is not required. Very simple illustrating examples prove that the criterion is satisfiable without. Clearly, it is recommended for practical applications to decompose the problem if the family is reducible. If an irreducible family has a spectral gap at 1, which is a necessary condition for termination of the algorithm suggested in [GP13], the existence of a $\mathscr{J}$-complete tree is guaranteed for a certain norm. Therewith, the set-valued tree method is in theory potentially functional in many cases which cannot be handled by certain extremal norm approaches.

Principally, all calculations of the method can be performed analytically or by means of interval arithmetic such that the strategy can be used for rigorous proofs. The latter would also allow a treatment of parametrized families, by subdividing the parameter interval into sufficiently small sub-intervals and finding for each of them a $\mathscr{J}$-complete tree that is valid for all contained parameter values.

To judge the practical value of the method, an algorithm was developed which bases on a variant of depth-first search on set-valued trees. A node of such a tree codes a set of matrix products, which is typically infinite. The backtrack criterion involves the check whether these matrix products are bounded by 1 with respect to some norm. The challenge was to compute an upper bound for the coded products which is on the one hand as close to the supremum as possible and on the other hand efficiently computable, leading to two different approaches. Both of them have certain requirements concerning the eigenvalues of the products, which restricts the algorithm in its present form to real matrix families.

In the smoothness analysis of linear subdivision schemes, JSR determination is needed to obtain Hölder regularity. This application initially motivated the thesis. The theoretical background for smoothness analysis of linear, univariate, stationary, uniform and compactly supported schemes with arbitrary arity is summarized and an explicit formula for the subdivision matrices is provided. It is shown that considering half of the tree is sufficient for certain norms in case of palindromic matrices. As another consequence of the symmetric situation, a palindromic family rarely possesses a spectral gap at 1. Families with a dominant generator pair of

length 2 are an exception and, as subdivision matrices, occur quite often[1]. To handle other cases, we present a transformation which, for a family with a dominant generator pair of odd length, leaves the JSR unchanged but enforces a spectral gap at 1. An appropriate transformation in case of generators with even length remains to be found.

The algorithm was implemented in MATLAB for families of real matrices in order to give a proof-of-concept. It proved to be useful for a range of examples, some illustrating the capabilities of the method for matrices with certain properties, others resulting from smoothness analysis of subdivision schemes. Numerical tests with high-dimensional or random matrices were not performed since the implementation is not optimized in terms of runtime and cannot compete with the impressive results of [GP13] in that respect.
However, long runtimes do not result necessarily from high-dimensional problems. An example from applications involved a pair of (10×10)-matrices and the computation was successful in sub-seconds, while termination might take days or weeks for certain lower-dimensional examples. The reason for difficulties in the computation should be analyzed systematically in future work. The algorithm apparently struggles in case of matrix products which are close to being an FP-product. So-called weak generators were introduced to handle these cases. Although this concept was successful for model problems, it seems to increase the computational effort a lot. Further problems may occur if the leading eigenvalue and the subdominant eigenvalue of an FP-product are very close in modulus, causing a slow decay of the subdominant eigenvalue of the powers with effect on the computational costs. Moreover, difficulties were observed in cases where an FP-product has a complex conjugate pair of leading eigenvalues. Possibly, the computed upper bound often is not close enough to detect a 1-bounded node in these cases such that an improvement of the bound estimation in case of complex eigenvalues is desirable.
The implementation involves different freely selectable parameters. In order to perform extensive test series, a framework should be built so that the computation is re-started with sophisticatedly changed parameter values if a test was not successful.

Although set-valued trees can be analyzed with respect to any submultiplicative norm, the choice of norm has an impact on the efficiency of the method. A combination of the set-valued tree approach and the extremal norm approach seems to be very promising. In case that the unit ball computation does not terminate after a certain number of iterations, some of these methods return an approximation of the JSR. If instead the computed norm is used for the set-valued tree approach, this possibly leads to the exact instead of an approximated value of the JSR for very short, and therewith efficiently computable, $\mathscr{J}$-complete trees. This might especially be interesting in cases of non-asymptotically simple families where the existence of certain extremal norms is not guaranteed.

[1] For example, palindromic (2×2)-matrices either satisfy $\hat{\rho}(\mathscr{A}) = \rho(A_1)$ or $\hat{\rho}(\mathscr{A}) = \rho(A_{[1,2]})^{\frac{1}{2}}$, see [Möß10]

Bibliography

[AS98] T. Ando and M. Shih. Simultaneous contractibility. *SIAM Journal on Matrix Analysis and Applications*, 19(2):487–498, 1998.

[BM02] T. Bousch and J. Mairesse. Asymptotic height optimization for topical IFS, tetris heaps, and the finiteness conjecture. *Journal of the American Mathematical Society*, 15(1):77–111, 2002.

[BN05] V.D. Blondel and Y. Nesterov. Computationally efficient approximations of the joint spectral radius. *SIAM J. Matrix Anal. Appl.*, 27(1):256–272, 2005.

[BNT05] V.D. Blondel, Y. Nesterov, and J. Theys. On the accuracy of the ellipsoid norm approximation of the joint spectral radius. *Linear Algebra and its Applications*, 394(0):91–107, 2005.

[BT00] V.D. Blondel and J.N. Tsitsiklis. The boundedness of all products of a pair of matrices is undecidable. *Systems and Control Letters*, 41(2):135–140, 2000.

[BTV03] V.D. Blondel, J. Theys, and A.A. Vladimirov. An elementary counterexample to the finiteness conjecture. *SIAM Journal on Matrix Analysis and Applications*, 24(4):963–970, 2003.

[BW92] M.A. Berger and Y. Wang. Bounded semigroups of matrices. *Linear Algebra and its Applications*, 166(0):21–27, 1992.

[BZ00] M. Broker and X.L. Zhou. Characterization of continuous, four-coefficient scaling functions via matrix spectral radius. *SIAM J. Matrix Anal. Appl.*, 22(1):242–257, 2000.

[CGSCZ10] A. Cicone, N. Guglielmi, S. Serra-Capizzano, and M. Zennaro. Finiteness property of pairs of 2×2 sign-matrices via real extremal polytope norms. *Linear Algebra and its Applications*, 432(2-3):796–816, 2010.

[CHR13] T.J. Cashman, K. Hormann, and U. Reif. Generalized Lane–Riesenfeld algorithms. *Computer Aided Geometric Design*, 30(4):398 – 409, 2013.

[DD89] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. *Constructive Approximation*, 5(1):49–68, 1989.

[DFH05] N. Dyn, M.S. Floater, and K. Hormann. A C^2 four-point subdivision scheme with fourth order accuracy and its extensions. *Analysis*, 1(128):f2, 2005.

[DL92a] I. Daubechies and J.C. Lagarias. Sets of matrices all infinite products of which converge. *Linear Algebra and its Applications*, 161:227–263, 1992.

[DL92b] I. Daubechies and J.C. Lagarias. Two-scale difference equations ii. local regularity, infinite products of matrices and fractals. *SIAM Journal on Mathematical Analysis*, 23(4):1031–1079, 1992.

[DL02] N. Dyn and D. Levin. Subdivision schemes in geometric modelling. *Acta Numerica*, 11:73–144, 2002.

[DLG87] N. Dyn, D. Levin, and J.A. Gregory. A 4-point interpolatory subdivision scheme for curve design. *Computer Aided Geometric Design*, 4:257–268, 1987.

[Dub86] S. Dubuc. Interpolation through an iterative scheme. *Journal of Mathematical Analysis and Applications*, 114(1):185–204, 1986.

[Els95] L. Elsner. The generalized spectral-radius theorem: An analytic-geometric proof. *Linear Algebra and its Applications*, 220:151–159, 1995.

[FM12] M.S. Floater and G. Muntingh. Exact regularity of pseudo-splines. *ArXiv e-prints*, September 2012.

[GMW94] T.N.T. Goodman, C.A. Micchelli, and J.D. Ward. Spectral radius formulas for subdivision operators. *Recent Advances in Wavelet Analysis*, 3:335–360, 1994.

[GP13] N. Guglielmi and V.Y. Protasov. Exact computation of joint spectral characteristics of linear operators. *Foundations of Computational Mathematics*, 13(1):37–97, 2013.

[Gri96] G. Gripenberg. Computing the joint spectral radius. *Linear Algebra and its Applications*, 234:43–60, 1996.

[GWZ05] N. Guglielmi, F. Wirth, and M. Zennaro. Complex polytope extremality results for families of matrices. *SIAM Journal on Matrix Analysis and Applications*, 27(3):721–743, 2005.

[GZ08] N. Guglielmi and M. Zennaro. An algorithm for finding extremal polytope norms of matrix families. *Linear Algebra and its Applications*, 428(10):2265–2282, 2008.

[GZ09] N. Guglielmi and M. Zennaro. Finding extremal complex polytope norms for families of real matrices. *SIAM J. Matrix Anal. Appl.*, 31(2):602–620, 2009.

[HIDS02] M.F. Hassan, I.P. Ivrissimitzis, N.A. Dodgson, and M.A. Sabin. An interpolating 4-point C^2 ternary stationary subdivision scheme. *Computer Aided Geometric Design*, 19(1):1–18, 2002.

[HMR09] J. Hechler, B. Mößner, and U. Reif. C^1-Continuity of the generalized four-point scheme. *Linear Algebra and its Applications*, 430(11-12):3019–3029, 2009.

[HMST11] K.G. Hare, I.D. Morris, N. Sidorov, and J. Theys. An explicit counterexample to the Lagarias-Wang finiteness conjecture. *Advances in Mathematics*, 226(6):4667–4701, 2011.

[HS08] K. Hormann and M.A. Sabin. A family of subdivision schemes with cubic precision. *Computer Aided Geometric Design*, 25(1):41–52, 2008.

[JCG14] R.M. Jungers, A. Cicone, and N. Guglielmi. Lifted polytope methods for computing the joint spectral radius. *SIAM Journal on Matrix Analysis and Applications*, 35(2):391–410, 2014.

[Jun09] R.M. Jungers. *The joint spectral radius, theory and applications*. Springer, 2009.

[Koz05] V. Kozyakin. A dynamical systems construction of a counterexample to the finiteness conjecture. In *Proceedings of the 44th IEEE Conference on Decision and Control and ECC 2005*, pages 2338–2343, 2005.

[LR80] J.M. Lane and R.F. Riesenfeld. A theoretical development for the computer generation and display of piecewise polynomial surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(1):35–46, 1980.

[LR94] R. Lima and M. Rahibe. Exact Lyapunov exponent for infinite products of random matrices. *Journal of Physics A: Mathematical and General*, 27(10):3427, 1994.

[LW95] J.C. Lagarias and Y. Wang. The finiteness conjecture for the generalized spectral radius of a set of matrices. *Linear Algebra and its Applications*, 214:17–42, 1995.

[Mae96] M. Maesumi. An efficient lower bound for the generalized spectral radius of a set of matrices. *Linear Algebra and its Applications*, 240:1–7, 1996.

[Mae00] M. Maesumi. Joint spectral radius and Hölder regularity of wavelets. *Computers and Mathematics with Applications*, 40(1):145–155, 2000.

[Mae08] M. Maesumi. Optimal norms and the computation of joint spectral radius of matrices. *Linear Algebra and its Applications*, 428(10):2324–2338, 2008.

[Möß10] B. Mößner. On the joint spectral radius of matrices of order 2 with equal spectral radius. *Advances in Computational Mathematics*, 33(2):243–254, 2010.

[MR14] C. Möller and U. Reif. A tree-based approach to joint spectral radius determination. *Linear Algebra and its Applications*, 463:154–170, 2014.

[PJ08] P.A. Parrilo and A. Jadbabaie. Approximation of the joint spectral radius using sum of squares. *Linear Algebra and its Applications*, 428(10):2385–2402, 2008.

[PJB10] V.Y. Protasov, R.M. Jungers, and V.D. Blondel. Joint spectral characteristics of matrices: A conic programming approach. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2146–2162, 2010.

[Pro96] V.Y. Protasov. The joint spectral radius and invariant sets of the several linear operators. *Fundamentalnaya i prikladnaya matematika*, 2(1):205–231, 1996.

[Pro05] V.Y. Protasov. The geometric approach for computing the joint spectral radius. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, pages 3001–3006, 2005.

[Rio92] O. Rioul. Simple regularity criteria for subdivision schemes. *SIAM Journal on Mathematical Analysis*, 23(6):1544–1576, 1992.

[RS60] G.C. Rota and W.G. Strang. A note on the joint spectral radius. *Indag. Math.*, 22(4):379–381, 1960.

[Sab10] M.A. Sabin. *Analysis and Design of Univariate Subdivision Schemes*. Geometry and Computing. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010.

[XY05] G. Xie and T. P.-Y. Yu. Smoothness analysis of nonlinear subdivision schemes of homogeneous and affine invariant type. *Constructive Approximation*, 22(2):219–254, 2005.

[ZZYZ07] H. Zheng, H. Zhao, Z. Ye, and M. Zhou. Differentiability of a 4-point ternary subdivision scheme and its application. *IAENG International Journal of Applied Mathematics*, 36(1):19–24, 2007.

Wissenschaftlicher Werdegang

Claudia Möller geb. Goltz
geboren am 01. Oktober 1984 in Aachen

2015	Promotion in Mathematik
2009 - 2015	Wissenschaftliche Mitarbeit und Promotionsstudium an der Technischen Universität Darmstadt
2009	Diplom in Mathematik
Sept. 2007 - Jan. 2008	Erasmus-Auslandssemester an der Université catholique de Louvain, Belgien
2004 - 2009	Studium der Mathematik mit Nebenfach Philosophie an der Technischen Universität Darmstadt
2004	Abitur am Gymnasium Marienschule in Hildesheim
1991 - 2004	Schulausbildung in Bonn und Hildesheim