On Simply Structured Bases of Graph Eigenspaces

Torsten Sander

Habilitationsschrift

vorgelegt bei der
Fakultät für Mathematik/Informatik und Maschinenbau
der Technischen Universität Clausthal

Oktober 2008

Contents

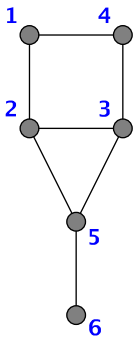
1 Introduction

As the name suggests, the discipline of algebraic graph theory connects graph theory and algebra — two areas of mathematics that at first sight could not be further apart from each other. Today, many synergies have been discovered. A graph theoretical problem may be represented in an algebraic way and then be solved by a seemingly unrelated algebraic theorem. Conversely, many algebraic problems are easier to represent and analyse if they can be phrased in the language of graph theory.

The formal foundation of algebraic graph theory has probably been laid in 1957 by the paper [15], written by Collatz and Sinogowitz. It defines the eigenvalues of graphs. Five years before, in 1952, Tutte had published a generalisation of his famous one-factorisation theorem [95]. It requires the solution of equations defined by the vertex-edge incidences and certain values assigned to graph vertices and edges. Several papers concerned with determinants and characteristic polynomials of matrices arising in mechanics, physics and chemistry had appeared even before that, cf. papers published by Rutherford [77] in 1947, by Pöschl and Collatz [73] in 1938, and by Funk [37] in 1935.

Probably the earliest contribution to algebraic graph theory has been made in 1931 by Hückel [52]. The molecular orbitals of electrons can be analysed by solving the Schrödinger equation. Hückel replaced this by a linear model and obtained a quantitative theory for the analysis of electron charges and locations (see e.g. [62] for an introduction). Essentially, it requires the determination of eigenvectors and eigenvalues of the molecular graph.

Being the earliest known link of algebraic graph theory, the paper [52] even today is still often quoted as a reference in papers analysing the spectral properties of graphs. Of course, today's chemists use modern computers to solve Schrödinger's equation numerically and mainly teach Hückel theory to students because it is easier to understand for undergraduates. Indeed, algebraic graph theory has found many more applications like biology, communication networks, computer graphics, tournaments, or internet search engines.



$$A(G) = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure 1.1: Example graph and its adjacency matrix

Let us shortly introduce the notion of eigenvalues and eigenvectors of graphs. The eigenvalues of a graph G with vertex set $V = \{v_1, \ldots, v_n\}$ are the eigenvalues of its adjacency matrix $A = (a_{ij})$ which is defined by $a_{ij} = 1$ if v_i is adjacent to v_j and $a_{ij} = 0$, otherwise. This eigenvalue definition is independent of the chosen vertex order. Note that there exist many more matrices whose spectra are of interest in algebraic graph theory. Moreover, the reader should be aware that algebraic graph theory is a vast research area in which the study of graph spectra only represents one of many ways in which graphs can be associated with algebraic structures.

We know that a given vector $w = (w_i)$ with $w \neq 0$ is an eigenvector of A for eigenvalue λ if and only if $Aw = \lambda w$. If we read this system of equations line by line and interpret w_i as a weight assigned to vertex v_i, then it is easily seen that, equivalently, for every vertex of G the sum over the values of its neighbours must equal λ times its own value. This is called the *summation rule*.

Consider the example graph in Figure 1.1. A basis of the kernel of this graph is shown in Figure 1.2.

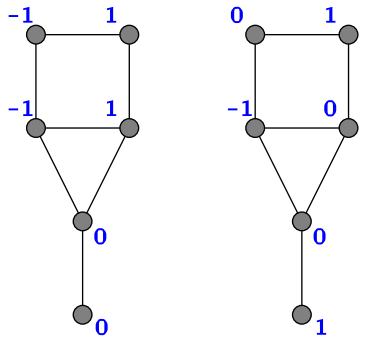


Figure 1.2: Example graph kernel basis

We can verify the validity of the vectors by applying the summation rule to all vertices. For example, the inner product of the second row of the adjacency matrix with a given vector asserts that the weight of the three neighbours of vertex 2 equals zero:

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{-1} \\ -1 \\ \mathbf{1} \\ 1 \\ \mathbf{0} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{0} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

The kernel of a graph and its dimension (called the *nullity* of the graph) are of particular interest to researchers of algebraic graph theory. There exist applications in chemistry ([52], [62], [90]), biology ([86]) and other areas.

Let us describe some of the more recent research on the kernel and nullity of graphs. Results on the nullity have been obtained for numerous graph classes. The nullity of bipartite graphs in general is treated in [19]. For trees it is known that the nullity is directly related to the size of a maximum matching. A tree is non-singular if and only if it has a perfect matching. The nullity of a tree equals the number of its vertices minus the number of vertices saturated by some maximum matching. These facts are very remarkable and have been proved several times, cf. [19], [9], [55], [35], [34], [78]. The construction of trees with prescribed number of vertices and maximum degree bound that achieve the maximum possible nullity is described in [34]. [35] gives a linear time algorithm for the computation of the nullity of a tree. In [60] these results are transferred to unicyclic graphs, yielding a linear time algorithm that can check whether a unicyclic graph is singular. In [97] it is shown that the nullity of a unicyclic graph on n vertices may range exactly between 0 and $n-4$. A general characterisation of graphs with maximum and near maximum nullity is achieved in [14]. The nullity has been determined for many more graph classes, e.g. for 4-circulants [23], line graphs of certain regular graphs [24], line graphs of trees [63], or distance powers of circuits [80].

A series of papers systematically investigates the nullity of graphs in terms of prototypical subgraphs [91], [87], [92], [89], [90]. The key concept is that the non-zero entries of graph eigenvectors induce certain subgraphs. Given a kernel eigenvector x, one partitions the graph into *periphery* and *core* vertices, depending on whether x vanishes on a vertex or not. The set of vertices that belong to some core is an invariant of the graph. A core graph is a graph with at least two vertices and some kernel eigenvector without zero entries. Core graphs with nullity one are called nut graphs. A minimal configuration is a graph of nullity one whose periphery is either empty or induces a subgraph of isolated vertices. Moreover, the removal of a peripheral vertex must increase the nullity of the graph. By allowing edges between peripheral vertices one obtains the notion of a singular configuration, which also has nullity one and contains a unique spanning minimal configuration as subgraph. The authors of the mentioned papers investigate the properties of these graphs and the roles they play as potential subgraphs of singular graphs. For example, a graph with nullity k contains k singular configurations as subgraphs such that their core vertices correspond to the non-zero entries of a kernel basis with an overall minimum number of non-zero entries [89]. These investigations have led to a complete characterisation of singular graphs in terms of singular configurations.

In the examples they present in their published work, researchers usually seek to give example eigenvectors that are as "simple" as possible. Since the adjacency matrix is integral one can find integral eigenspace bases for all integer eigenvalues. But often enough the eigenvector entries are chosen such that as few different numbers as possible are used. For example, one can find numerous examples where eigenvectors with entries only from the set $\{0, 1, -1\}$ are used and even complete eigenspace bases consisting only of such vectors, cf. [74], [55], [66], [89], [90] and Figure 1.2. This may seem a coincidence, but it is well worth investigating which graph classes and eigenvalues admit such eigenvectors and bases, which we shall call *simply structured*.

The case of finding particularly simple eigenspace bases is a rather new research topic. Owing to a certain degree of sparsity, a simply structured basis can have significant computational advantages. Depending on the graph class, simply structured bases may have storage advantages. For example, the vectors of a simply structured basis of a tree kernel can be compressed into bit fields by simply discarding the entry signs, because it is possible to reconstruct the distribution of signs [78]. Finally, it is noted in [90] that $\{0, 1, -1\}$-eigenvectors of certain molecular graphs are related to equidistributivity of electron charges in non-bonding molecular orbits.

It has long been known that the vector $(1, 1, \ldots, 1)^T$ spans the eigenspace for eigenvalue k of a connected k-regular graph and that $(1, -1, 1, -1, \ldots)^T$ spans the eigenspace for eigenvalue $-k$ of a connected bipartite k-regular graph. In [10] a construction is given that allows the generation of a simple kernel vector for each pair of vertices with identical neighbourhoods. Moreover, it is easily verified that the complete graph K_n with eigenvalues -1 and $n-1$ admits simply structured eigenspace bases for both eigenvalues.

Apart from this folklore, the first serious research in the direction of simply structured bases has probably been undertaken by Villarreal. In a rather algebraic context it is shown in the 1995 paper [96] that the kernel of the *incidence matrix* of a graph always admits a basis with entries from the set $\{-2, -1, 0, 1, 2\}$. Recall that the incidence matrix of an undirected graphs is a $\{0, 1\}$ valued matrix that captures the edge-to-vertex incidence relation, just like the adjacency matrix reflects the vertex-to-vertex adjacency relation. For two-connected graphs (i.e. graphs without cut-vertices) it was shown in 2002 that the set $\{-2, -1, 0, 1, 2\}$ can be reduced to $\{-1, 0, 1\}$ [49]. This even holds for graphs without cut-edges [6]. It is even possible to characterise all graphs whose incidence matrix admits a simply structured kernel basis, effectively exhausting this direction of research.

In contrast, the situation is much harder for the adjacency matrix of a graph. Current research has not progressed as far as to yield a complete characterisation of graphs with simply structured bases. In 2004, a proof for the existence of a simply structured basis for the kernel of a tree with nullity one was given in the thesis [79]. However, it is noted that the proof technique used inevitably fails for greater nullity. One year later the result was extended to all singular trees in [78]. Independently and probably unaware of this reference, the existence of simply structured tree kernels has later been also shown in two other papers [5], [66]. In [82] a characterisation of all unicyclic graphs admitting simply structured bases is achieved. Moreover, in [58] it is proved that all eigenspaces of unitary Cayley graphs admit simply structured bases.

In the following chapters we will present numerous results on simply structured eigenspace bases. To begin with, a few general criteria will be proved. However, it becomes clear that much more powerful results can be obtained when the focus is restricted to some specific graph class. In sections 4 and 5 we study the eigenspaces of trees and unicyclic graphs, respectively. In section 6 we treat distance powers of paths and circuits. Section 7 deals with graph products and related classes,

namely Cayley graphs, Hamming graphs and Sudoku graphs. The eigenspaces for eigenvalues 0 and -1 of the graphs without an induced path on four vertices, called cographs, are investigated in section 8. Finally, section 9 is concerned with the kernels of coprime graphs. These graphs model a number theoretic problem initially proposed by Erdős.

2 Basics

In this chapter we introduce basic notation and definitions. Further notation is postponed until later and introduced when required by the context. Moreover, we state basic results that may be used implicitly in subsequent chapters.

Matrices and vectors

The symbols $\mathcal{I}$, $\mathcal{J}$ and j denote the identity matrix, the all ones matrix and the all ones vector, respectively. Subscripts may be used to indicate their respective dimensions in case they are not clear from the context. Let $\operatorname{rk} M$, $\operatorname{im} M$, $\ker M$ denote the rank, image and kernel, respectively, of a given matrix M.

Given row or column vectors $v_1, \ldots, v_k$ such that $v_i^{(1)}, \ldots, v_i^{(n)}$ are the entries of v_i, we define their concatenation (which is a row vector) by

$$(v_1 \mid v_2 \mid \ldots \mid v_k) := (v_1^{(1)}, \ldots, v_1^{(n)}, \ldots, v_k^{(1)}, \ldots, v_k^{(n)}).$$

A matrix in which the i-th column vector can be derived from the first column vector by means of a downward rotation by $i-1$ entries is called a circulant matrix [25].

Let us abbreviate $\omega_n = e^{\frac{2\pi \mathrm{i}}{n}}$.

Theorem 2.1. [10] Let $(a_1, a_2, \ldots, a_n)^T$ be the first column of a real circulant matrix A. Then the eigenvalues of A are exactly

$$\lambda_r = \sum_{j=1}^{n} a_j \omega_n^{(j-1)r}, \quad r = 0, \ldots, n-1.$$

□

Graphs

Throughout this work — unless stated otherwise — we only consider finite, loopless, simple graphs.

Note that this section does not replace a thorough introduction to graph theory, for which the reader is referred to e.g. [46], [28], [53], [57]. The foundations of algebraic graph theory can be found in sources like [10], [17], [18], [39].

Let G be a graph. Then by $V(G)$ and $E(G)$ we denote its sets of vertices and edges, respectively. Let $\deg_G(x)$ denote the degree of vertex x in graph G. We write $x \sim y$ if the vertices x, y are adjacent in the considered graph.

We write $N_G(x)$ for the neighbourhood of the vertex x in G, i.e. the set of all vertices of G that are adjacent to x.

Given a set $M \subseteq V(G)$, we denote by $G - M$ the graph formed by removing the vertices of M and all their adjacent edges from G. If only a single vertex x is removed, then for the sake of convenience we write $G - x$ instead of $G - \{x\}$. If H is a subgraph of G, then $G - H := G - V(H)$.

We define two join operations, namely the disjoint union $G \cup H$ and the complete join $G \triangledown H$ of the vertex disjoint graphs G and H. The first operation simply joins the vertex and edge sets whereas the latter additionally adds all possible edges between the two joined vertex sets.

Let K_n, P_n, C_n denote the complete graph, path and circuit on n vertices, respectively.

By a matching of a graph we understand a set of independent edges of the graph. We call a vertex covered or saturated by a matching if it is incident to an edge of that matching. A maximum matching comprises the greatest possible number of edges. Maximum matchings of trees are unique.

By $\overline{G}$ we denote the complement of the graph G.

A graph is called bipartite if there exists a disjoint partition of its vertex set into two sets such that every edge of the graph runs from a vertex of the first set to a vertex of the second set. We shall call such a partition a corresponding vertex bipartition of the bipartite graph.

A circulant graph is a graph whose adjacency matrix is circulant with respect to a suitable vertex ordering. Examples for circulant graphs are circuits and their distance powers.

Graph eigenspaces

Let $G = (V, E)$ be a graph with vertex set $V = \{x_1, \ldots, x_n\}$.

Then its adjacency matrix $A(G) = (a_{ij})$ is defined by

$$a_{ij} = \begin{cases} 1 & \text{if } x_i \sim x_j \\ 0 & \text{else} \end{cases}.$$

Note that $A(\overline{G}) = \mathcal{J} - A(G) - \mathcal{I}$.

The eigenvalues of a graph G are the eigenvalues of its adjacency matrix $A(G)$. Note that this definition is invariant under isomorphism so that the eigenvalues of a graph do not depend on the numbering of its nodes.

An eigenvector x of a graph G for eigenvalue λ is given by a non-zero solution of the equation $A(G)x = \lambda x$. One can interpret the eigenvectors of $A(G)$ as vertex weight functions $V \to \mathbb{R}$, in order to derive a notion of graph eigenvectors that does not depend on the numbering of the vertices. Let $V(G) = \{v_1, \ldots, v_n\}$ and $x = (x_1, \ldots, x_n)^T$. Now assign value x_i to vertex v_i. Reading the equation $A(G)x = \lambda x$ for each entry of x separately, it is easily seen that, equivalently, for every vertex of G the sum over the values of its neighbours must equal λ times its own value:

$$\sum_{v_j \in N(v_i)} w(v_j) = \lambda w(v_i) \qquad (i = 1, \ldots, n).$$

We will hereafter refer to this as the *summation rule*.

The eigenspace for eigenvalue λ of graph G is written as $\mathrm{Eig}(\lambda; G) := \ker(A(G) - \lambda\mathcal{I})$. Moreover, $\ker G := \mathrm{Eig}(0; G)$ is the kernel of G. By $\mu(\lambda; G) := \dim \mathrm{Eig}(\lambda; G)$ we denote the multiplicity of the eigenvalue λ of graph G. Clearly, $0 \leq \mu(\lambda; G) \leq |V(G)|$. We say that a G is singular if it has a zero eigenvalue. The multiplicity of the zero eigenvalue is called the nullity of a graph.

Since adjacency matrices are symmetric we can immediately deduce that the eigenvalues of a graph are all real numbers and that its eigenspaces are mutually perpendicular.

For a bipartite graph λ is an eigenvalue if and only if $-\lambda$ is an eigenvalue of the graph as well, with the same multiplicity [10]. Given an eigenvector for eigenvalue λ of a bipartite graph, one obtains an eigenvector for $-\lambda$ by negating the entries of the vector on all vertices of exactly one of the two sets of a corresponding vertex bipartition.

An important consequence of the famous Perron-Frobenius theorem on non-negative matrices [38] is that the maximum eigenvalue λ_{max} of a connected graph G with at least to vertices has multiplicity one and is the only eigenvalue to afford a positive eigenvector.

The largest eigenvalue of a k-regular graph is $\lambda_{max} = k$. The corresponding eigenspace is spanned by the all-ones vector j.

Theorem 2.2. [10] Let G be a k-regular graph on s vertices and let $\lambda_1, \ldots, \lambda_s$ be its eigenvalues.

Then the eigenvalues of $\overline{G}$ are $s - k - 1$ and all numbers $-\lambda_i - 1$ where $\lambda_i \neq k$. The eigenspace for eigenvalue $-\lambda_i - 1$ of $\overline{G}$ is the same as the eigenspace for eigenvalue $\lambda_i \neq k$ of G. The eigenspace for eigenvalue $s - k - 1$ is spanned by the all ones vector. □

As a consequence of Theorem 2.1, we can explicitly state the spectrum of circulant graphs:

Theorem 2.3. Let G be a circulant graph. Let $(0, a_2, \ldots, a_n)^T$ be the first column of a circulant adjacency matrix of G. Then the eigenvalues of G are exactly

$$\lambda_r = \sum_{j=2}^{n} a_j \omega_n^{(j-1)r}, \quad r = 0, \ldots, n-1.$$

□

According to [25], the column vectors of the matrix

$$F^* = n^{-\frac{1}{2}} \left(\omega_n^{(i-1)(j-1)}\right)_{i,j=1,\ldots,n} \in \mathbb{C}^{n \times n},$$

which is the conjugate transpose of the so-called Fourier matrix $F \in \mathbb{C}^{n \times n}$, constitute a complete and universal set of complex eigenvectors for *every* circulant matrix M of order n. Moreover, the $(r+1)$-th column of F^* yields a complex eigenvector for eigenvalue λ_r of Theorem 2.3.

3 General criteria

In this chapter we derive general criteria which eigenvalues of a graph potentially admit simply structured eigenspace bases. Naturally, such eigenvalues must necessarily be integers.

Theorem 3.1. Let G be a graph with a pendant vertex. If G admits a $\{0, 1, -1\}$ valued eigenvector v for eigenvalue λ and v does not vanish on all pendant vertices of G and their neighbours, then $\lambda \in \{-1, 0, 1\}$. □

Proof. Let x be a pendant vertex of G and v a $\{0, 1, -1\}$ valued eigenvector for eigenvalue λ such that $w(x) \neq 0$. From the summation rule for vertex x we derive the condition $\lambda = w(y)/w(x)$. Since $w(x), w(y) \in \{-1, 0, 1\}$ this condition can only be satisfied if $\lambda \in \{-1, 0, 1\}$. ■

Corollary 3.2. Let G be a tree. If G admits a $\{0, 1, -1\}$ valued eigenvector for eigenvalue λ, then necessarily $\lambda \in \{-1, 0, 1\}$. □

Proof. It suffices to observe that no eigenvector of a tree can vanish on all of its leaves, as can be shown by a simple inductive argument. ■

The next theorem is particularly useful for graphs where the neighbourhoods of adjacent vertices overlap to a large extent, e.g. for distance powers (cf. section 6).

Theorem 3.3. Let G be a graph and T a spanning tree of G such that there exists $q \in \mathbb{N}$ with $|(N(x) \cup N(y)) \setminus (N(x) \cap N(y))| \leq q$ for all adjacent vertices x, y of T.

If v is a $\{-1, 0, 1\}$ valued eigenvector of G for eigenvalue λ, then $\lambda \in \{-q+1, \ldots, q-3\}$ or $v = j$. □

Proof. Consider the summation rule on G for two adjacent vertices x, y of T:

$$\lambda w(x) = \sum_{z \in N(x)} w(z) = \sum_{z \in N(x) \setminus N(y)} w(z) + \sum_{z \in N(x) \cap N(y)} w(z),$$
$$\lambda w(y) = \sum_{z \in N(y)} w(z) = \sum_{z \in N(y) \setminus N(x)} w(z) + \sum_{z \in N(x) \cap N(y)} w(z).$$

We find that

$$(\lambda + 1)(w(x) - w(y)) = \sum_{z \in N(x) \setminus (N(y) \cup \{y\})} w(z) - \sum_{z \in N(y) \setminus (N(x) \cup \{x\})} w(z)$$
$$\in \{-(q-2), \ldots, (q-2)\}.$$

We have $w(x) - w(y) \in \{-2, -1, 0, 1, 2\}$ so that, consequently, $w(x) = w(y)$ or $\lambda \in \{-q+1, \ldots, q-3\}$. The proof is completed by iterating the argument along incident edges of T. ■

We call a matrix A totally unimodular if every square submatrix $\tilde{A}$ of A has $\det(\tilde{A}) \in \{0, 1, -1\}$.

Theorem 3.4. For every totally unimodular, singular matrix A there exists a simply structured basis of $\ker(A)$. □

Proof. Let $\mathrm{rk}(A) = r$. We arrange the lines and columns of A so that the first r lines and the first r columns form an invertible submatrix A'. If n is the number of columns in A, then the dimension of $\ker(G)$ is $p = n - r$. To solve $Ax = 0$ with $x = (x_i) \in \mathbb{R}^n$ we may take arbitrary values for $x_{r+1}, \ldots, x_n$ and determine $x_1, \ldots, x_r$ by the first r equations in $Ax = 0$. We define basis vectors $x^{(k)} = (x_i^{(k)})$ by $x_{r+j}^{(k)} = \delta_{k,j}$ for $j = 1, \ldots, n-r$ and $k = 1, \ldots, n-r = p$. If $\tilde{x}^{(k)} = (x_1^{(k)}, \ldots, x_r^{(k)})$, then we have $A'\tilde{x}^{(k)} = -c_k$, where c_k is the $(r+k)$-th column of A reduced to positions $1, \ldots, r$. We solve $A'\tilde{x}^{(k)} = -c_k$ by Cramer's rule, which means

$$x_i^{(k)} \;=\; -\det(A_i)/\det(A') \qquad (i = 1, \ldots, r).$$

Here A_i is obtained from A' by replacing column i by c_k. Both matrices A' and A_i are submatrices of A. As A is totally unimodular, their determinants have values 0 or ± 1 and so $x_i^{(k)} \in \{0, 1, -1\}$. ■

A direct consequence of Theorem 3.4 is that every graph with totally unimodular adjacency matrix admits a simply structured kernel basis.

4 Trees

The spectra of trees have been investigated quite exhaustively. For example, trees are exactly those graphs whose characteristic polynomial equals the so-called matching polynomial, whose coefficients are derived from the numbers of matchings of different sizes [40]. The square of the maximum eigenvalue of a tree lies roughly between one and two times the maximum degree [40]. The inverse of the adjacency matrix of a non-singular tree has the extraordinary property that its zero-nonzero pattern corresponds to the adjacency matrix of a graph that can be directly constructed from the given tree itself [41],[69]. The paper [43] analyses the relative signs of eigenvectors on adjacent vertices of a tree. Many more results could be quoted here.

In this chapter we give a characterisation of trees for which simply structured eigenspace bases exist. According to Corollary 3.2, the only feasible eigenvalues that allow the construction of such bases are $0, 1, -1$.

The next two sections 4.1 and 4.2 deal with simply structured kernel bases. In [79] the existence of a similarly structured basis has been proved for the kernel of a simply singular tree. However, the proof technique used inevitably fails for multiply singular trees. We will use an alternate approach that works for arbitrary singular trees. Based on the observation that straight application of the Gauss algorithm to the adjacency matrix of a tree often yields a kernel basis of the intended structure, we seek to develop a procedure that guarantees the construction of such a basis. The key is a possibly incomplete Gaussian elimination scheme whose pivoting strategy only considers unit vector rows of the coefficient matrix. In section 4.4 the focus is extended to other elimination procedures.

In section 4.5 a partitioning technique is developed that is used in section 4.6 to characterise all trees whose eigenspaces for eigenvalue 1 or -1 admit simply structured bases.

4.1 The FOX algorithm

To construct the kernel basis of a graph we usually perform Gaussian elimination on the adjacency matrix of the given graph. Since we want to construct a basis that exhibits certain properties there will typically arise additional rules to observe.

In this section we present an elimination algorithm that performs a possibly incomplete Gaussian elimination scheme. Its pivoting strategy only considers unit vector rows of the coefficient matrix. Applied to an adjacency matrix, this only turns some nonzero entries into zero entries.

For a formal analysis we want to avoid excessive use of matrix indices and therefore seek a presentation of the algorithm that carries out the elimination operations directly on a directed graph. If an undirected graph is given, then this approach

uses the directed graph with the same adjacency matrix as the given undirected graph. The elimination procedure described below only removes edges from the digraph.

Let $G = (V, E)$ be a connected graph. Construct the corresponding bidirectional orientation $\hat{G}$, i.e. the unique digraph that has the same adjacency matrix as G.

We will introduce an algorithm that takes the graph $\hat{G}$ as input and produces a labelled subgraph $\tilde{D}$ of the input graph. Each node will be labelled with at least one of the three labels `F`, `O` and `X`.

Algorithm 4.1.

(1) Let all nodes of $\hat{G}$ be unlabelled.

(2) Find a node v without `X` label that has exactly one outgoing edge e. If no such node exists, go to step (6).

(3) Tag node v with `X`.

(4) Let w be the unique node with $e = vw$. Except for e, remove all incoming edges of w.

(5) Tag node w with `O` and go to step (2).

(6) Tag all unlabelled nodes with `F`.

□

Since for each run of the main loop from step (2) to (5) another node becomes `X` labelled, it is clear that the algorithm stops after at most $|V|$ iterations of the main loop.

Let $H_{\{\text{labels}\}}$ be the subgraph of $\tilde{D}$ induced by all nodes that have each been labelled with all of the labels mentioned in the subscript.

We will say that v is an `O` labelled node if among its labels there is an `O` label. Conversely, we require an $\overline{\texttt{O}}$ labelled node to have no `O` label.

It is obvious that every node of $\tilde{D}$ has been labelled.

An example is shown in figure 4.1, where the original graph G is a tree. The final graph $\tilde{D}$ shown in the figure is obtained for the step (2) node sequence 0, 2, 4, 5, 13, 14, 17, 20, 22, 21.

We will now look for further properties of $\tilde{D}$. The construction of `X` and `O` labelled nodes immediately implies

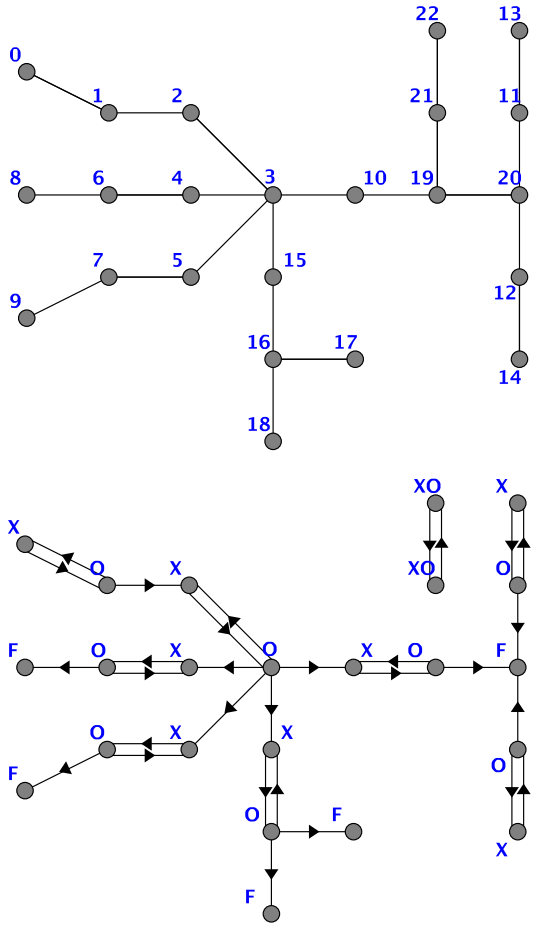


Figure 4.1: Example graphs G and $\tilde{D}$

Lemma 4.2. [78] Every X labelled node of $\tilde{D}$ has exactly one outgoing neighbour, namely an O labelled node. Every O labelled node of $\tilde{D}$ has exactly one incoming neighbour, namely an X labelled node. Every node has at least one incoming neighbour. □

Lemma 4.3. [78] Every weak component of $H_{\mathtt{XO}}$ is a 2-cycle and also a weak component of $\tilde{D}$. □

Proof. Let v_1 be an XO labelled node. By Lemma 4.2 we have exactly one outgoing edge $e = v_1v_0$. The neighbour v_0 is O labelled. Now suppose that $\tilde{D}$ does not contain the edge v_0v_1. Then, by Lemma 4.2, for v_1 there exists an incoming edge v_2v_1 from an X labelled node v_2. Since v_2v_1 is an edge of $\tilde{D}$ but not v_0v_1 we see that $v_2 \neq v_0$. But v_2 misses v_1v_2 as incoming edge since v_1v_0 is the only outgoing edge of the X labelled node v_1. Consequently, v_2 must be an O labelled node.

Repeat this reasoning for v_2 instead of v_1 and so on. We obtain a sequence of distinct XO labelled nodes $v_1, v_2, \ldots$ with v_i adjacent to v_{i-1} for $i \geq 1$. Naturally, this sequence is finite. Note that the final node v_n may only be O labelled by v_0 since this is the only node left without an X label towards the end of constructing the node sequence.

As a consequence, there exists a cycle of XO labelled nodes in $\tilde{D}$. But in $\hat{G}$ the nodes of this cycle originally each had at least two outgoing neighbours. Therefore, one of the O labelled nodes of the cycle must originate from an exterior X labelled node. But, by construction, each node of the cycle has been O labelled exactly by one of its neighbours within the cycle, a contradiction.

Therefore, there exists an edge v_0v_1 in $\tilde{D}$. By Lemma 4.2 we see that v_0 and v_1 form a 2-cycle in $\tilde{D}$. ■

Lemma 4.4. [78] Let v_1 be an F labelled node of $\tilde{D}$ and v_1v_2 be an outgoing edge of v_1. Then v_2 is an F labelled node as well and v_2v_1 is also an edge of $\tilde{D}$. □

Proof. Let $e = v_1v_2$. Then v_2 cannot be an O labelled node since otherwise the edge e would have been removed in $\tilde{D}$. Also, v_2v_1 is an edge in $\tilde{D}$ because v_1 is not an O labelled node and therefore may not miss any of its original incoming edges. Consequently, v_1 and v_2 are mutually adjacent in $\tilde{D}$.

Clearly, v_2 cannot be X labelled because its only outgoing neighbour v_1 would have to be an O labelled node. Therefore, v_2 must be F labelled. ■

Lemma 4.5. [78] Let v be an F labelled node and let H be the strong component of $\tilde{D}$ that contains v. Then all adjacencies between nodes of H are mutual. H contains only F labelled nodes. Further, $|H| = 1$ or $|H| \geq 3$. In the second case H contains a bidirectional cycle with at least 3 nodes.

Incoming neighbours of H can only be $\overline{\text{XO}}$ labelled nodes. H has no outgoing neighbours. □

Proof. Let w be a node of H, $w \neq v$. Then there exists a directed path from v to w. By Lemma 4.4 this path contains only F labelled nodes. Therefore, the entire strong component H contains only F labelled nodes, in particular w. By Lemma 4.4 it also follows that all adjacencies of nodes of H are mutual.

It is clear that H cannot have outgoing edges since by Lemma 4.4 these would only lead to F labelled nodes which cannot be weakly connected to H. Conversely, incoming edges cannot start from F labelled nodes but neither from X labelled nodes

since that would require H to contain an O labelled node. The only remaining alternative is an $\overline{\text{X}}$O labelled node.

Assume $|H| = 2$. Then each node would only have exactly one outgoing edge and should have been X labelled, a contradiction.

Now let $|H| \geq 3$. Then H contains two mutually adjacent nodes v_0, v_1. Since v_1 is not an X labelled node it must have another outgoing edge v_1v_2 besides v_1v_0. Note that $v_2 \neq v_1$ must be an F labelled node. Repeat this conclusion for v_2 and so on until a maximal sequence of distinct nodes $v_0, v_1, \ldots, v_n$ has been constructed. Since H is finite we see that v_n must be adjacent to one of the nodes $v_0, \ldots, v_{n-2}$. Thus, H contains a bidirectional cycle with at least 3 nodes. ■

Corollary 4.6. [78] Let G be a tree. Then H_{F} contains only isolated points. □

Lemma 4.7. [78] Assume that H_{F} contains only isolated points.

Then the F labelled nodes of $\tilde{D}$ are exactly the $\overline{\text{X}}$ labelled nodes that have no outgoing edges. Incoming edges of F labelled nodes always start from $\overline{\text{X}}$O labelled nodes. □

Proof. Let v be an $\overline{\text{X}}$ labelled node of $\tilde{D}$ that has no outgoing edges and assume that v is also O labelled. Then there exists an X labelled incoming neighbour w. Since by assumption there exists no edge vw in $\tilde{D}$ we see that w is missing an incoming edge an therefore must be O labelled. But v lies in the same weak component of $\tilde{D}$ as w so that by Lemma 4.3 there must be an X label on v, a contradiction. Therefore v must be an F labelled node. The rest follows from Lemma 4.5. ■

Lemma 4.8. [78] If the FOX algorithm is conducted in a way that avoids the assignment of multiple labels for as long as possible, then the set of unlabelled nodes at the time of the first assignment of a second label to an already labelled node is identical to the set of nodes that become F labelled at the end of the FOX algorithm. □

Proof. We need to show that the assignment of a second label to a node does not affect the set of unlabelled nodes.

The only situation where a second label can be assigned is when there exists an O labelled node y with only one outgoing neighbour x. Note that x must be the X partner of y. By construction, all other neighbours of x must be incoming neighbours. These are necessarily O labelled since their incoming edge from x is missing.

After the FOX algorithm has assigned an X label to y and an O label to x we see that x and y form a separate 2-cycle. Therefore, if among the other formerly incoming neighbours there is one with only one outgoing neighbour, then the situation is the same as before and we may proceed in the same manner. Overall, only nodes of type X or O are affected. ■

Let us now turn to some immediate algebraic properties of the FOX algorithm. Since the FOX algorithm only performs row operations of the adjacency matrix of the given graph G, it is obvious that it has the same kernel as the adjacency matrix of the resulting graph $\tilde{D}$:

In the following, let D denote the undirected counterpart of $\tilde{D}$.

Lemma 4.9. [82] For any graph G,

$$\ker G = \ker \tilde{D} = \ker D$$

□

Proof. The equality $\ker G = \ker \tilde{D}$ directly follows from the construction of $\tilde{D}$. Now consider the FOX run on G whose result is $\tilde{D}$. Let $e = xy$ be an edge that lies in G but not in the subgraph D. Then according to the definition of the FOX algorithm e can only join O labelled nodes. Further, according to Lemma 4.2 the X partner of x cannot be y and vice versa. It follows that the considered FOX run on G could be performed in the same way even if the edge e did not exist. Consequently, there exists a FOX run on D that yields $\tilde{D}$. ■

Every X labelled node has exactly one outgoing neighbour, therefore its corresponding row of $A(\tilde{D})$ is a unit vector. Consequently, the value of the O labelled partner of this X labelled node must be zero for every vector from the kernel of G:

Lemma 4.10. [78] Let G be a graph. Then the nodes of $H_{\mathtt{O}}$ obtained after a run of the FOX algorithm on G form a subset of the nodes whose value is zero for every vector from the kernel of G. □

We will see that this lemma can be strengthened considerably if the FOX algorithm is applied to a tree or forest.

4.2 FOX on trees

In this section we apply the FOX algorithm from section 4.1 to trees. There already exist other algorithms that operate directly on a tree in order to determine spectral properties. For example, in [54] an algorithm is presented that computes the

determinant of the neighbourhood matrix (which is the sum of adjacency matrix and identity matrix $\mathcal{I}$) of a tree. In [35] the algorithm is rephrased such that it computes the characteristic polynomial of a tree. It can also test whether a tree has a certain prescribed eigenvalue. An algorithm that can compute eigenspace bases of trees is described in [55]. However, that algorithm still requires the solution of a linear equation system as its final step, cf. section 4.4. In contrast, we will describe a construction for a simply structured tree kernel basis that works entirely on graphs.

In the following, we will assume that given a graph G we will automatically apply the FOX Algorithm 4.1 to its bidirectional orientation $\hat{G}$ to get the final digraph $\tilde{D}$. The number of nodes of G and the number of $\mathtt{F}$ labelled nodes of $\tilde{D}$ will be denoted by $n = |G|$ and $k = |H_{\mathtt{F}}|$, respectively.

We may combine the properties of the FOX algorithm from section 4.1 in a straightforward manner to obtain the following lemma. Note that we call a directed path *maximal* if it is not contained in any other directed path.

Lemma 4.11. [78] Let G be a tree. For any $\mathtt{X}\overline{\mathtt{O}}$ labelled node v that has only one incoming edge let $\tilde{R}_v$ be the subgraph of $\tilde{D}$ spanned by the nodes of all directed paths in $\tilde{D}$ starting from v. Likewise, for any $\mathtt{F}$ labelled node w let $\tilde{S}_w$ be the subgraph of $\tilde{D}$ spanned by the nodes of all directed paths in $\tilde{D}$ ending at w. The undirected counterparts of $\tilde{R}_v$ and $\tilde{S}_w$ in G are called R_v and S_w, respectively.

Then:

1. The graphs R_v and S_w each induce a subtree of G.
2. Each maximal directed path in $\tilde{R}_v$ that begins at v starts with an even number of consecutive nodes labelled $\mathtt{X}\overline{\mathtt{O}}$ and $\overline{\mathtt{X}}\mathtt{O}$ alternatively and terminates with an $\mathtt{F}$ labelled node.
3. Each maximal directed path in $\tilde{S}_w$ that leads to w starts with an even number of consecutive nodes labelled $\mathtt{X}\overline{\mathtt{O}}$ and $\overline{\mathtt{X}}\mathtt{O}$ alternatively and terminates with an $\mathtt{F}$ labelled node.

□

Before we advance to the construction of kernel eigenvectors, we will investigate some consequences of the previous lemma and definitions.

Lemma 4.12. [78] Let G be a tree. Then for every $\mathtt{X}\overline{\mathtt{O}}$ labelled node v of $\tilde{D}$ there exists a node $w \in H_{\mathtt{F}}$ of $\tilde{D}$ such that v lies in $\tilde{S}_w$. □

Proof. Consider the last node w of a directed path P of maximum length within $\tilde{D}$ that leads away from v. Let x be the immediate predecessor of w along P. We show that w can only be `F` labelled. Then by definition v would lie in $\tilde{S}_w$.

Assume that w is `O` labelled. Then w cannot have any neighbours in $\tilde{D}$ besides x. But then w and x must be of type `XO`. Consequently, either $v = x$ so that it must also be of type `XO` or there exists no directed path from v to w because of Lemma 4.3.

Now assume that w is `X` labelled. Then due to the maximality of P the only outgoing neighbour of w must be x. Hence, x is the `O` partner of w. The case $v = x$ is impossible because then v would be `O` labelled. But since x is `O` labelled it cannot have an incoming edge from the third last node along P. ■

Lemma 4.13. [78] Let G be a tree and x an `F` labelled node of G. Then, all $\overline{\texttt{XO}}$ labelled nodes of S_x have degree 2 within this subgraph of G. Consequently, all nodes of S_x whose degree is at least 3 are necessarily of type $\texttt{X}\overline{\texttt{O}}$ or `F`. □

Proof. Clearly, S_x cannot contain any `XO` labelled nodes. Let u be a type $\overline{\texttt{XO}}$ node of S_x and let v be the `X` partner of u. Since v has no outgoing neighbours in $\tilde{S}_x$ besides u it follows that a directed path from v to the `F` labelled node of $\tilde{S}_x$ must lead via u. But then u must have another outgoing neighbour w besides v. It is impossible for u to have yet another outgoing neighbour y because then the `F` labelled node could not be reached in $\tilde{S}_x$ without introducing a cycle in the tree G. Since u has no incoming neighbours besides v we see that it has degree two in S_x. ■

The next lemma states that within a subgraph S_w any $\overline{\texttt{O}}$ labelled node may play the part of the `F` labelled node.

Lemma 4.14. [82] Let G be a tree and w a node that gets `F` labelled by a given FOX run $\mathcal{A}$. Further, let v be an $\texttt{X}\overline{\texttt{O}}$ labelled node of S_w.

Then there exists a FOX run $\mathcal{B}$ on G that creates the same node labels as run $\mathcal{A}$, except that node v gets `F` labelled and w becomes $\texttt{X}\overline{\texttt{O}}$ labelled. Moreover, the subgraphs $S_w^{(\mathcal{A})}$ and $S_v^{(\mathcal{B})}$ of G are identical. □

Proof. We may assume w.l.o.g. that $\mathrm{dist}_G(v, w) = 2$, otherwise iterate the proof technique.

According to Lemma 4.11 the node u that lies between v and w is `O` labelled. Clearly, u must be the partner of v. Let $u_1, \ldots, u_r$ be the other incoming neighbours of w in $\tilde{D}$, all being type `O`.

Since w is F labelled we see that the operations carried out by FOX run $\mathcal{A}$ may be rearranged such that first only the branches that are connected to w via $u_1, \ldots, u_r$ get labelled and then only the branch connected via u. This does not change the result $\tilde{D}$ or any node labels. Now, when the FOX algorithm would select v we pick w instead. This is possible because the neighbours $u_1, \ldots, u_r$ have all been O labelled so that its only outgoing neighbour is u. Since v was eligible for X labelling one step before and the O labelling of u has removed the edge vu we see that v now becomes F labelled. Denote this altered FOX run by $\mathcal{B}$.

Obviously, except for the edges between the nodes u, v, w and their labels there is no difference between the results of runs $\mathcal{A}$ and $\mathcal{B}$. Since there exits a directed path from w to v in $\tilde{D}^{(\mathcal{B})}$ and u has no other incoming edges other than from w it is clear that $S_w^{(\mathcal{A})}$ and $S_v^{(\mathcal{B})}$ are the identical subgraphs of G. ■

We will now exhibit several properties of the adjacency matrix of the graph $\tilde{D}$ obtained for a tree.

Lemma 4.15. [78] Let G be a tree and let A be the adjacency matrix of $\tilde{D}$. Then the row vectors of A that correspond to the $\overline{\texttt{XO}}$ labelled of $\tilde{D}$ are linearly independent among themselves and from the remaining row vectors of A. □

Proof. Let H be the subgraph of $\tilde{D}$ that is spanned by all $\texttt{X}\overline{\texttt{O}}$ labelled and all $\overline{\texttt{XO}}$ labelled nodes. For each X labelled node of H remove the edge leading back from its unique outgoing neighbour. Then, as a consequence of Lemma 4.11 it is possible to assign integral height levels to the nodes of H such that the edges starting at each level may only lead to the nodes of the level directly below.

Construct a node sequence $v_1, \ldots, v_n$ as follows. First number the X labelled nodes of H level by level, starting from the top level. For the same order of the X labelled nodes, number the corresponding outgoing O labelled neighbours. Finally, number the XO labelled and then the F labelled nodes.

Assume that the adjacency matrix A reflects the above node numbering. Now form the submatrix $M = (m_{ij})$ of A that contains only the rows that correspond to the $\overline{\texttt{XO}}$ labelled nodes of $\tilde{D}$ and the columns that correspond to the $\texttt{X}\overline{\texttt{O}}$ labelled nodes. Then by construction we have $m_{ij} = 0$ if $i > j$ and $m_{ii} = 1$. Thus, M takes upper triangular form with an all ones diagonal and therefore has maximum rank. Since $\texttt{X}\overline{\texttt{O}}$ labelled nodes can only have $\overline{\texttt{XO}}$ labelled nodes as incoming neighbours we see that within those columns of A that correspond to the $\texttt{X}\overline{\texttt{O}}$ labelled nodes all non-vanishing entries must necessarily belong to M as well. Consequently, the result follows. ■

Construction 4.16. [78] Let G be a tree. Assign weight 0 to all `O` labelled nodes of $\tilde{D}$ and also to all `F` labelled nodes except one node w. Assign weight 1 to w and construct $\tilde{S}_w$ according to Lemma 4.11.

All `X`$\overline{\texttt{O}}$ labelled nodes that are not contained in $\tilde{S}_w$ receive weights of 0. Conducting an incoming edge breadth first search on $\tilde{S}_w$ with starting node w, we can assign weights to the remaining `X` labelled nodes as follows. Let x be an unweighted `X` labelled node to be processed and let y be its outgoing `O` labelled neighbour. Then assign to x the negative sum of the weights of all non-incoming neighbours of y.

Performing this procedure for every `F` labelled node of $\tilde{D}$ we obtain a set of k vectors $z_i \in \mathbb{R}^n$. □

Observation 4.17. [78] The following properties of Construction 4.16 are obvious:

1. At any time, all non-incoming neighbours of y have already been assigned weights.
2. The k vectors z_i are linearly independent.

□

Lemma 4.18. [78] The vectors obtained by Construction 4.16 have only entries from the set $\{0, 1, -1\}$. □

Proof. Use the notation from Construction 4.16 and assume that we need to assign a weight to the `X` labelled node x. Consider the non-incoming neighbours of the associated node y. These can only be either `X`$\overline{\texttt{O}}$ labelled or `F` labelled. A non-zero weight can only occur at the neighbour that lies on the unique directed path from y to w. Using an inductive argument it is clear that only weights 1 or -1 can be assigned to `X` labelled nodes along this path and, consequently, to the node x. ■

For a forest we will now strengthen Lemma 4.10 from the previous section.

Lemma 4.19. [78] Let G be a forest. Then the graph $H_{\texttt{O}}$ contains exactly those nodes for which every vector from the graph kernel vanishes. In particular, this node set is invariant for all possible results of the FOX algorithm. □

Proof. Consider Construction 4.16 and note that according to Lemma 4.12 every `X`$\overline{\texttt{O}}$ or `F` labelled node v lies in a subgraph $\tilde{S}_w$ for some node w. We see that for at least one basis vector the construction yields a nonzero weight on v. ■

Example 4.20. For graphs that contain cycles it is possible to find counterexamples regarding the above lemma. For instance, consider the graph that results by connecting a fifth node to one of the nodes of the cycle C_4. Its kernel has dimension one and the spanning vector is zero on an X labelled node, see figure 4.2.

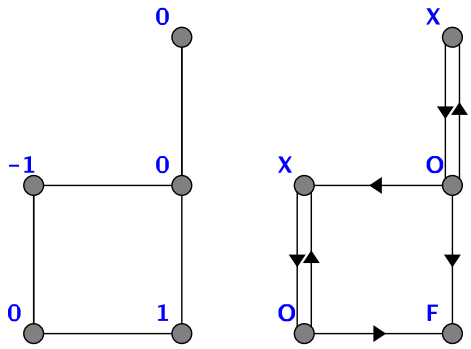


Figure 4.2: Counterexample for Lemma 4.19

□

Next, we present our main theorem. It is useful to note that a tree has a perfect matching if and only if it is nonsingular since the rank of the adjacency matrix of a tree is twice the size of a maximum matching [19], [55]. By iterative construction of the matching starting from the leaves and removing matched node pairs from the graph it becomes clear that a perfect matching in a tree must be unique.

Theorem 4.21. [78] Let G be a tree. Apply Algorithm 4.1 on its bidirectional orientation $\hat{G}$ to get the final digraph $\tilde{D}$.

Then $k = |H_{\mathtt{F}}|$ equals the degree of singularity of G.

For $k \geq 1$ Construction 4.16 yields a basis of $\ker G$ that consists of vectors with entries from $\{0, 1, -1\}$ only.

For $k = 0$ the graph $\tilde{D}$ consists only of XO labelled nodes. The perfect matching of G can be constructed by pairing the nodes according to the 2-cycles in $H_{\mathtt{XO}}$. □

Proof. Construct row and column index permutations for the adjacency matrix as follows. First take all row indices that correspond to XO labelled nodes and arrange them as pairs according to the 2-cycles of $\tilde{D}$. Use the same order of the node pairs for the column indices, but swap the node indices of each pair. Next, append the indices of the $\mathtt{X}\overline{\mathtt{O}}$ labelled nodes to the row indices and then in the same order append the indices of the corresponding unique outgoing ($\overline{\mathtt{X}}\mathtt{O}$ labelled) neighbours to the column indices. Now repeat this procedure vice versa (swap row and column indices) and

additionally observe the node order described in the proof of Lemma 4.15. Finally append the indices of the F labelled nodes both to the row and column indices.

Using the above index permutation, the adjacency matrix of $\tilde{D}$ transforms into the matrix

$$\begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & R & * \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

with an upper right triangular matrix R. This follows directly from Lemmas 4.3, 4.2, 4.5, 4.7, and 4.15.

Observing Lemma 4.15, we see that this matrix has kernel dimension k. Further, it has the same rank as the adjacency matrix of G since Algorithm 4.1, essentially, simply repeatedly looks for a unit vector row and uses it for pivoting.

Case $k \geq 1$. Since the vectors constructed are linearly independent and Lemma 4.18 asserts that their entries are as required, it suffices to show that these vectors belong to the kernel of G. Using each vector as node weights on G, this is equivalent to showing that for each node the weight sum of its neighbours vanishes. Observe that two nodes that are adjacent in G are disconnected in $\tilde{D}$ if and only if both have been O labelled. Therefore we may consider $\tilde{D}$ instead (to be precise, its simple undirected representation). The neighbours of an F labelled or X labelled node are all O labelled nodes, therefore the weight sum vanishes. An $\overline{\text{X}}$O labelled node may have both X labelled and F labelled neighbours. Among the latter there is at most one non-vanishing weight. Referring to Construction 4.16 we see that this weight has exactly been chosen so that the weight sum vanishes.

Case $k = 0$. First observe that by pairing each X labelled node with its unique outgoing (O labelled) neighbour a perfect matching of G can be constructed since there are no F labelled nodes. As a consequence, every directed path in $\tilde{D}$ that connects two leaves of G must contain an even number of nodes. Next, note that every O labelled leaf of $\tilde{D}$ must also be X labelled since it has exactly one outgoing neighbour. Since the latter becomes O labelled this pair of nodes forms a 2-cycle. Repeat this argument for the remaining nodes. It is impossible to have only X$\overline{\text{O}}$ labelled leaves left since this would mean that we could find a directed leaf-to-leaf path with an odd number of nodes. Therefore, the whole graph $\tilde{D}$ falls into XO labelled 2-cycles (cf. Lemma 4.3) so that indeed a perfect matching of G is obtained. ■

Corollary 4.22. [19], [9], [55], [35], [34], [78] The rank of the adjacency matrix of a tree is twice the size of a maximum matching. □

Proof. A matching that misses only k nodes can be constructed by pairing each X labelled node with its unique outgoing neighbour.

Conversely, consider a maximum matching of G. Without changing the size of the matching we may alter it such that every leaf neighbour is paired with one of its leaves. Apply Algorithm 4.1 and begin with the leaves that belong to the matching. These will become `X` labelled and their neighbours receive `O` labels. The subgraph of G that is spanned by the remaining unlabelled nodes is a forest F. Alter the matching of G such that every leaf neighbour of F is paired with one of its leaves. Proceed by alternatingly continuing Algorithm 4.1 with the leaves of F that belong to the matching and updating F. Finally, the forest F will only contain isolated nodes because otherwise the matching would not be maximal. Clearly, exactly the nodes of F become `F` labelled by Algorithm 4.1. ■

As Corollary 4.22 indicates, there exists a connection between the FOX algorithm and maximum matchings on trees. If fact, the following results show that there exists a one-to-one mapping between the possible outcomes of applying the FOX algorithm to a tree and the maximum matchings of that tree.

Lemma 4.23. [82] Let M be a maximum matching of the tree T. Then M covers at least one leaf of T. □

Proof. Assume to the contrary that no leaves are covered by M. Let v_1 be such a leaf and v_2 its neighbour. Then v_2 must be covered because M is maximum. Let v_3 be the neighbour of v_2 with $v_2v_3 \in M$. By assumption v_3 cannot be a leaf. Therefore v_3 has another neighbour v_4 with $v_4 \neq v_2$. v_4 cannot be covered by M because otherwise $v_1, \ldots, v_4$ would form an augmenting path, contradicting the choice of M. Now the situation for v_4 is like it was for v_2. Continue the argument. Since T is a finite graph, we arrive at a contradiction. ■

Lemma 4.24. [82] Let M be a maximum matching of the tree T. Then there exists a FOX run on T such that M can be obtained by collecting the edges between the `X` labelled nodes and their `O` labelled partners. □

Proof. Let T_i be a subtree of T and let $M_i = M \cap E(T_i)$ be the restriction of M to T_i. Let the directed graph $\tilde{T}^{(i)}$ be a valid intermediate graph of a FOX run on T. Further, let $\tilde{T}_i$ be the restriction of $\tilde{T}^{(i)}$ to T_i (including labels).

Now we make the following assumptions:

(a) M_i is a maximum matching on T_i.

(b) None of the nodes of $\tilde{T}_i$ have been labelled.

(c) The only nodes of $\tilde{T}^{(i)}$ eligible for `X` labelling lie in the subgraph $\tilde{T}_i$.

(d) Every node eligible for `X` labelling in $\tilde{T}_i$ is also eligible in $\tilde{T}^{(i)}$.

(e) In $\tilde{T}^{(i)}$ the edges between the `X` labelled nodes and their `O` labelled partners form the set $M - M_i$.

(f) None of the unlabelled nodes of $\tilde{T}^{(i)} - \tilde{T}_i$ are covered by M.

By Lemma 4.23 and assumption (a) there exists a leaf $v_1^{(i)}$ of T_i that is covered by M_i. Let $v_2^{(i)}$ be its neighbour in T_i. Clearly, the edge $e = v_1^{(i)} v_2^{(i)}$ belongs to M_i. We conclude from assumptions (b), (c) and (d) that $v_1^{(i)}$ is eligible for `X` labelling, both in $\tilde{T}_i$ and $\tilde{T}^{(i)}$. In both graphs we can perform a step of the FOX algorithm that labels $v_1^{(i)}$ with `X` and then $v_2^{(i)}$ with `O`. From $\tilde{T}^{(i)}$ we obtain the new intermediate graph $\tilde{T}^{(i+1)}$.

Let $N_i = \{v_3^{(i)}, \ldots, v_{s_i}^{(i)}\}$ be the neighbours of $v_2^{(i)}$ besides $v_1^{(i)}$. We can deduce from assumptions (c) and (d) that none of the nodes of N_i have any outgoing edges in $\tilde{T}^{(i+1)}$, hence we already know they will become `F` labelled at the end of the FOX run. The nodes of N_i cannot be covered by M because of assumptions (e) and (f). Now construct the subtree T_{i+1} of T by removing the nodes $v_1^{(i)}, \ldots, v_{s_i}^{(i)}$ from T_i. Note that $M_{i+1} = M_i - e$. Incrementing i by one, it is straightforward to see that assumptions (a) to (f) hold again.

Starting with $T_1 = T$ and $M_1 = M$ we see that assumptions (a) to (f) hold for $i = 1$. Iterate the induction step described above until $M_i = \emptyset$. Complete the FOX run on $\tilde{T}^{(i)}$. It is clear by construction that FOX will only assign `F` labels to the remaining unlabelled nodes, which concludes the proof. ■

Lemma 4.25. [82] Let M be the set of edges between the `X` labelled nodes and their `O` labelled partners obtained by a FOX run on a given tree T. Then M is a maximum matching on T. □

Proof. Obviously M is a matching on T. Exactly the `F` labelled nodes are not covered by M. In view of Theorem 4.21 and Corollary 4.22 this means that indeed M is a maximum matching. ■

Corollary 4.26. [82] Exactly the `O` labelled nodes of a tree are covered by every single maximum matching. □

Proof. This follows from Lemmas 4.12, 4.14, 4.24, and 4.25. ■

Remark 4.27. The subgraphs S_w associated with the vectors obtained by Construction 4.16 are actually minimal configurations (cf. section 1). To see this we check the defining properties of a minimal configuration. Each of the basis vectors corresponds to a different subgraph S_w. Clearly, a maximum matching of such a graph S_w misses exactly one vertex, so it has nullity one. Construction 4.16 guarantees that its periphery consists only of isolated vertices (which are exactly the `O` labelled vertices). Moreover, the removal of a peripheral vertex increases the nullity of the graph. This follows from the fact that the removal splits the graph into exactly two components (a tree S_w cannot have an `O` labelled leaf). The `X` partner of the removed vertex cannot lie in the same component as the one containing the `F` labelled vertex. It follows easily that on each of the components a maximum matching would miss a single vertex.

Trivially, the trees S_w are even singular configurations. Recall that in section 1 we cited a theorem that every graph with nullity k contains k singular configurations as subgraphs such that their core vertices correspond to the non-zero entries of a kernel basis with an overall minimum number of non-zero entries [89]. It is not difficult to see that a FOX kernel basis has this minimal basis property so that the k trees S_w used to construct the basis are examples for the singular configurations predicted by the mentioned theorem. □

We conclude this section with the following extension lemma:

Lemma 4.28. [78] Let G be a tree and v be an `O` labelled node of G. Construct a tree G' by connecting v to a new node w.

Setting the weight of w to zero it is then possible to embed every vector from the kernel of G such that a vector from the kernel of G' is obtained.

Moreover,

$$\dim\ker G' = \dim\ker G + 1$$

and there exists a basis of $\ker G'$ such that all but one vector have zero weights on w. □

Proof. Since according to Lemma 4.19 all vectors from the kernel of G have a zero entry on node v it is readily checked that assuming a zero weight on w will extend any eigenvector from $\ker G$ to one from $\ker G'$. Conversely, every vector from $\ker G'$ that has weight zero on w can be reduced to a vector from $\ker G$ by deleting the entry corresponding to w. Hence, $\dim\ker G \leq \dim\ker G'$. Moreover, we can find a basis of $\ker G'$ that does not contain more than one vector with a nonzero weight on w since a suitable linear combination of two such vectors would have a zero weight on w and therefore be a vector extended from $\ker G$. Consequently, $\dim\ker G' \leq \dim\ker G + 1$.

Now run the FOX Algorithm 4.1 on G as described in Lemma 4.8. The operations that occur until the first multiple label is assigned are also valid for G'. But then v would be `O` labelled on G' so that w would become `F` labelled at the end of the FOX run. In view of Lemma 4.8 we see that we have gained exactly one degree of freedom and may construct a vector from $\ker G'$ with a nonzero weight on w. ■

4.3 Storing FOX vectors

Given a basis whose vectors have length n and contain only entries from the set $\{-1, 0, 1\}$, it is clear that on a computer these vectors can be stored very efficiently. By separating the positive and negative components we get two binary vectors so that the overall information of a basis vector can be stored in a bit field of length $2n$.

Considering a tree kernel basis with entries only from the set $\{-1, 0, 1\}$, it is even sufficient to save only the zero/non-zero pattern of these basis vectors. Using a recursive technique it is possible to adjust the signs of the entries such that from every bit field a proper kernel vector can be reconstructed. However, the following example shows that the reconstructed vectors need not be linearly independent.

Example 4.29. Consider the tree $K_{1,4}$. The leftmost three pictures in figure 4.3 represent a kernel basis of $K_{1,4}$. Given the zero/non-zero patterns of these basis vectors (e.g. read from suitable bit fields in memory), clearly, the first two basis vectors can be reconstructed up to a sign factor. Reconstruction of the third vector from its bit field may, however, also yield the vector shown in the rightmost picture. Since this vector can be obtained by subtracting the first basis vector from the second, we may fail to reconstruct a kernel basis from the bit fields of the original basis vectors.

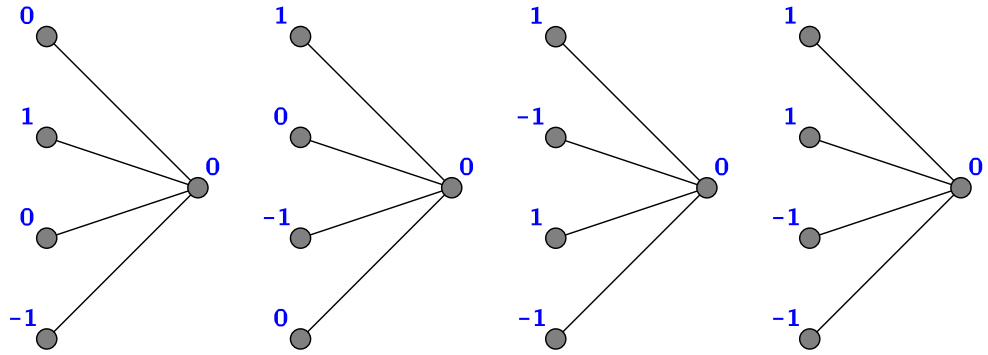


Figure 4.3: Kernel basis reconstruction for $K_{1,4}$

□

The next theorem asserts for a tree kernel basis obtained by the FOX algorithm and Construction 4.16 that up to sign factors the entire basis can be reconstructed from the stored zero/nonzero patterns.

Theorem 4.30. [78] Let G be a tree with n nodes. Having stored the zero-nonzero pattern of a vector from the kernel of G obtained by Construction 4.16 in a bit field of length n, the original eigenvector may be reconstructed up to a factor of ± 1 by the following procedure:

1. Interpreting the bit field as a function on the nodes of G, let G' be the tree that is obtained by removing all branches from G that have only zero weights.

2. Choose a node w of G' with weight 1 and negate the weights of all nodes whose distance to w is $2, 6, 10$, and so on.

□

Proof. First note that in view of Construction 4.16 the nonzero entries of the bit field represent $\overline{\texttt{O}}$ tagged nodes of G. Hence, node w must be either `F` or $\texttt{X}\overline{\texttt{O}}$ tagged. According to Lemma 4.11 and Construction 4.16 the graph G' is one of the subtrees S_x of G in which w lies according to Lemma 4.12.

In view of this, we see that the neighbours (within G) of a nonzero node of G' must be `O` tagged so that their weight sum vanishes. Now we need to check the neighbour weight sums for `O` tagged nodes. Note that all node weights outside G' are zero. By definition of G' any node z outside G' can only have one neighbour in G'. This neighbour in turn must be `O` tagged or otherwise the neighbour weight sum for z could not vanish. It remains to check the neighbour weight sums for the nodes of G'. But since all node weights outside G' are zero we may entirely restrict ourselves to the subgraph G'. According to Lemma 4.13 all `O` tagged nodes of G' have degree 2 in G'. But by construction these neighbours have weights 1 and -1.

We see that we have constructed an eigenvector from $\ker G$ with the same zero/nonzero pattern as the original vector. Moreover, Lemma 4.13 ensures that after assigning only one nonzero weight to a node there is no choice during the reconstruction of the remaining entries. ■

Remark 4.31. Note that although in the proof of the previous theorem we rely on the node tags created by the FOX algorithm, these tags are not needed for the eigenvector reconstruction itself. □

4.4 Gaussian elimination

We have shown that the FOX algorithm can be used to construct a simply structured kernel bases for any given singular tree. A simple experiment suggests that it may be far easier to construct simply structured tree kernel bases. Just use a computer algebra system like "Maple" to determine the kernel of a tree adjacency matrix. Surprisingly, "Maple" always presents a simply structured basis. The following theorem explains this observation:

Theorem 4.32. If a kernel basis of a tree adjacency matrix is computed by Gaussian elimination where for backward substitution the free variables are successively chosen such that that the corresponding entries of the resulting vectors are unit vectors, then this kernel basis is simply structured. □

Proof. Observe that the adjacency matrix of a tree is totally unimodular, as can be shown by a simple induction argument. The result now follows from the proof of Theorem 3.4. Just note that the values chosen for the free variables completely determine the resulting solution vector. The corresponding entries are given the same values as in Theorem 3.4, assuming suitable re-indexing. Note that re-indexing of vertices changes the adjacency matrix but not its property of total unimodularity.

■

Note that Theorem 4.32 does not obsolete Theorem 4.21 since it does not help understand the structure of the computed bases nor hint at the close connection to matchings.

In the paper [55] by John and Schild, an algorithm is presented that permits the computation of tree eigenvectors by means of successively assigning vectors to all vertices of an augmented tree, collecting the vectors assigned to certain vertices and then solving a linear system of usually much smaller dimension than the one arising from the adjacency matrix itself. The authors present an example tree (cf. Figure 4.4) for which the computed basis of the kernel is strictly simple (cf. Figure 4.5). We show that this is not a coincidence, but that, under normal conditions, the John-Schild algorithm always generates a simply structured tree kernel basis.

Let $\mathrm{vec}(x)$ denote the vector assigned to vertex x of a given graph. Then the John-Schild algorithm to determine the eigenvectors for eigenvalue λ of a tree T with n vertices works as follows:

1. Determine a *complete path system* (CPS) for the given tree T. This is a spanning subgraph of T whose components are paths.

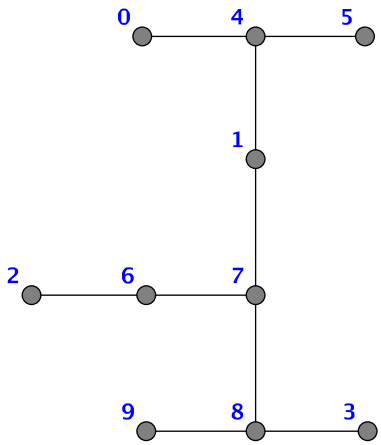


Figure 4.4: Example tree from John-Schild paper

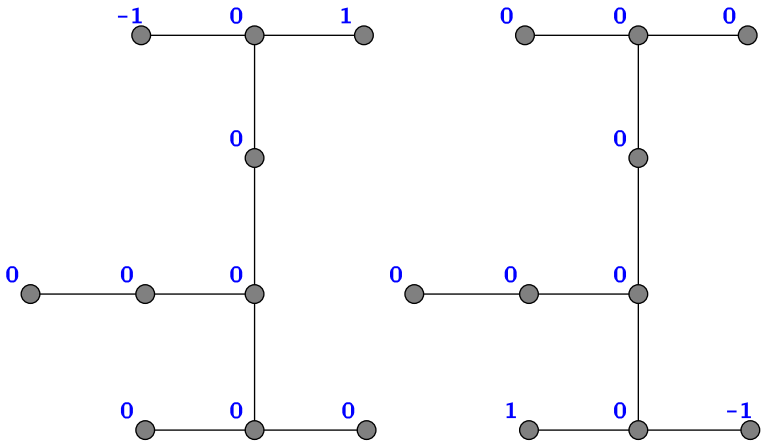


Figure 4.5: Kernel basis for example tree from John-Schild paper

2. Orient every path of the CPS from one of its terminal vertices to the other (from *start vertex* to *end vertex*). Transfer these orientations to T to obtain a partially oriented tree T'.

3. To every vertex of T' that coincides with a path end vertex attach an outgoing edge to a newly introduced *sink* vertex (whereas the path start vertex is called a *source* vertex).

4. Assign distinct unit vectors of dimension k to the source vertices (k being the number of source vertices).

5. Iteratively propagate vectors along the paths as follows: Choose a vertex v for which all neighbours of its unique predecessor vertex w except v have been assigned vectors. Assign to v the vector $\lambda\,\mathrm{vec}(w) - \sum \mathrm{vec}(x)$ where the sum ranges over all neighbours of w except v.

6. Number the vertices of T as $v_1, \ldots, v_n$ such that $v_1, \ldots, v_k$ are the source vertices. Form the $(n \times k)$-matrix D whose rows are $\mathrm{vec}(v_1)^T, \ldots, \mathrm{vec}(v_n)^T$. Number the sinks of T' as $v'_1, \ldots, v'_k$ such that v'_i is the sink that corresponds to source v_i. Form the $(k \times k)$-matrix D^* whose rows are $\mathrm{vec}(v'_1)^T, \ldots, \mathrm{vec}(v'_k)^T$.

7. Compute a kernel basis $B = \{b_1, \ldots, b_q\}$ of D^*. Then an eigenspace basis of T for eigenvalue λ is given by the set $\{Db_1, \ldots, Db_n\}$.

Note that the eigenvalue λ can be treated symbolically so that the vector propagation only needs to be carried out once for all eigenvalues.

Theorem 4.33. The John-Schild algorithm always computes a simply structured tree kernel basis, provided that for the computation of the basis B the free variables of the system $D^*z = 0$ are assigned values in unit vector fashion. □

Proof. Note that the first k rows of D form an identity matrix. Hence, the product Db_i contains a copy of b_i in its first k entries. Thus we can treat the vectors of B as if we had obtained them by directly solving the system $A(T)z = 0$ and assigning the free variables in unit vector fashion. So the result follows from Theorem 3.4 in the same way as Corollary 4.32. ■

4.5 Partitioning trees by eigenvectors

Since the eigenvectors of a graph can be considered as real valued functions on its vertex set, one may partition the vertices of a tree by grouping those vertices on which there exist non-zero values for some vector from a fixed eigenspace and those on which every vector from that space vanishes. Then the components arising from vertices of the first kind can be contracted into single vertices. Together with any adjacent vertices they induce a so-called skeleton forest, a concept initially hinted at in [67]. We show that the null space of this skeleton provides a blue print for the vectors from the considered eigenspace of the original tree. Moreover, its matching properties can be utilised to determine the eigenspace dimension. A converse technique allows to create trees with certain eigenspace properties from given skeletons.

In section 4.2 we have seen how easily the matching properties of a tree help determine a kernel basis. Depending on whether a vertex can be saturated by some maximum matching or not, we know if there exists a kernel eigenvector that does not vanish on that vertex. The overall picture here is that, for eigenvalues potentially different from 0 we can derive a similar relationship by means of the skeleton forest. Whenever a vertex of the skeleton can be saturated by a maximum matching we can find an eigenvector for the original graph that is completely non-zero on the vertices contracted into the considered skeleton vertex.

The partitioning of graphs according to eigenvector structure has been studied before, but with different aims. For example, in [74] the components induced by the signs of the entries of eigenvectors have been investigated and an upper bound for the number of zero entries in eigenvectors of graphs has been derived.

Associate with a given real $n \times n$ symmetric matrix (so-called pattern matrix) an undirected graph $\Gamma(A)$ on n vertices $1, 2, \ldots, n$ by including the edge connecting vertex i to vertex j in the edge set if and only if the entry at position (i, j) in A is non-zero. In 1998, Nylen [67] showed that if $\Gamma(A)$ is a tree, then the dimension of the null space of A equals the number of connected components of the subgraph of $\Gamma(A)$ induced by the set of indices i such that x_i is non-zero for some vectors x from the null space A, minus the number of indices adjacent to (with respect to $\Gamma(A)$) but not in that index set. This also follows from and is even generalised by the results presented below since they are readily extended to pattern matrices.

Tree pattern matrices are also called *acyclic matrices*. A number of results exist on the spectra of such matrices in general. One of the first papers in that area was [33], where it is shown that the signs of eigenvector entries are related to the position of the corresponding eigenvalue in the ordered spectrum of a tree pattern matrix. In [56] it is shown that it is in general not possible to express maximum eigenvalue multiplicities of a class of acyclic matrices given by some tree in terms of the degrees of the vertices. It is moreover well known that any tree T has at least $\mathrm{diam}(T) + 1$ distinct eigenvalues. This fact can be generalised to tree pattern matrices [59]. For even more general results that cover conditions on the solution of linear equations or existence of certain eigenvalues and eigenvectors for a class of matrices whose zero-nonzero pattern matches a given digraph, see e.g. [64].

4.5.1 Tree eigenvector decomposition

Let G be a graph and $M = \{X_1, \ldots, X_r\}$ a set of mutually vertex disjoint subgraphs of G. Then by $G/\{X_i\}_{i=1}^r$ or G/M we denote the graph that results from the contraction of each subgraph X_i in G to a single vertex x_i. Further, let $\mathfrak{C}(G)$ denote the set of components of G.

Let x be an eigenvector for eigenvalue λ of graph G. Let $N_\lambda(G, x)$ be the set of those vertices of G on which x vanishes. Moreover, let $N_\lambda(G)$ mean the set of vertices on which every eigenvector for eigenvalue λ of G vanishes.

In the sequel, we will investigate the properties of the sets $N_\lambda(G, x)$ and $N_\lambda(G)$. But first we cite the following result due to Fiedler that will required in several proofs:

Lemma 4.34. [33] Let T be a tree. Let v be an eigenvector of T for eigenvalue λ. If v does not have any zero entries, then λ necessarily has multiplicity one. □

Lemma 4.35. [83] Let G be a graph and x an eigenvector for its eigenvalue λ. Then:

1. For any $C \in \mathfrak{C}(G - N_\lambda(G,x))$, the restriction $x|_C$ is an eigenvector of the graph C for eigenvalue λ. If G is a tree, then $x|_C$ constitutes an eigenspace basis of the subtree C for eigenvalue λ.

2. For any $C \in \mathfrak{C}(G - N_\lambda(G))$ the restriction $x|_C$ is either the null vector or an eigenvector of the graph C for eigenvalue λ. If G is a tree and $x|_C \neq 0$, then $x|_C$ does not contain any zero entries and, moreover, constitutes an eigenspace basis of the subtree C for eigenvalue λ.

□

Proof. In the first case the claim follows directly from the definition of $N_\lambda(G,x)$, the summation rule and Lemma 4.34. The case $C \in \mathfrak{C}(G - N_\lambda(G))$ is similar, with only one additional argument. Let $v_1, \ldots, v_k$ be the vertices of C. For every vertex v_i there exists an eigenvector x_i of T for eigenvalue λ whose restriction $x_i|_C$ does not vanish on v_i. It is straightforward to show (see e.g. Lemma 7 in [67]) that there exists a linear combination of these vectors x_i that has no zero entries so that by Lemma 4.34 the associated eigenvalue λ of C has multiplicity one. ■

It is easy to see that the eigenspace basis related claim of the lemma does not extend to general graphs. For example, look at the complete graph K_3 and its eigenvectors $(2,-1,-1)$, $(-1,2,-1)$ for eigenvalue -1.

Let x be an eigenvector for eigenvalue λ of a given tree T. Further let C_i, $i = 1, \ldots, r$, be the elements of $\mathfrak{C}(T - N_\lambda(T,x))$. We will now concentrate on a particularly interesting subset of $N_\lambda(T,x)$. Namely, let $N_\lambda^C(T,x)$ consist of all those vertices of $N_\lambda(T,x)$ that are adjacent to at least one of the subgraphs C_i in T.

Lemma 4.36. [83] Let T be a tree and x an eigenvector for its eigenvalue λ. Let C_i, $i = 1, \ldots, r$, be the elements of $\mathfrak{C}(T - N_\lambda(T,x))$. Further let c_i denote the associated contracted vertices of $T/\{C_i\}_{i=1}^r$.

Then the vertex set $N_\lambda^C(T,x) \cup \{c_1, \ldots, c_r\}$ induces a forest F in $T/\{C_i\}_{i=1}^r$ such that the leaves of F form a subset of $\{c_1, \ldots, c_r\}$ and are also leaves of $T/\{C_i\}_{i=1}^r$. □

Proof. Clearly, the contraction $T/\{C_i\}_{i=1}^r$ of the tree T by the sub-forest $\bigcup\{C_i\}$ is a tree. So the induced subgraph F of $T/\{C_i\}_{i=1}^r$ must be a forest.

Now, consider an element v of $N_\lambda^C(T,x)$. By construction and since T is a tree there exists a one-to-one mapping of the non-zero weight neighbours of v to a subset of $\mathfrak{C}(T - N_\lambda(T,x))$. By definition, v is adjacent to at least one component C_i, but since the sum over the neighbours of v must vanish we see that it must be adjacent to at least two such components. Consequently, v is adjacent to at least two of the vertices c_i in both $T/\{C_i\}_{i=1}^r$ and F. So the leaves in F are a subset of $\{c_1,\ldots,c_r\}$.

Assume that c_k is a leaf of F that is not a leaf of $T/\{C_i\}_{i=1}^r$. Then in $T/\{C_i\}_{i=1}^r$, there would exist a neighbour w of c_k such that $w \in N_\lambda(T,x) \setminus N_\lambda^C(T,x)$. But then w could only be adjacent to zero-weight vertices, a contradiction. ■

In the following, let $S_\lambda(T,x)$ denote the forest F by Lemma 4.36 associated with a given tree T and eigenvector x. We call $S_\lambda(T,x)$ the *x-skeleton* of T.

Note that x-skeletons do not necessarily characterise an eigenspace basis, i.e. there may exist linearly independent eigenvectors x, x' for eigenvalue λ of a tree T such that $\mathfrak{C}(T - N_\lambda(T,x)) = \mathfrak{C}(T - N_\lambda(T,x'))$. An example is shown in Figure 4.6 for $\lambda = 1$.

Figure 4.6: Eigenvectors with the same x-skeleton

Remark 4.37. In [87], [91] graphs with nullity one and corresponding eigenvector without zero entries, so-called nut graphs, are studied. Although the components of $T - N_0(T,x)$ have nullity one and corresponding eigenvector without zero entries, the theory on nut graphs does not yield any insight in the case of trees since the respective components are all isomorphic to K_1 and nut graphs, however, are required to have at least two vertices. Recalling the results from section 4.2 and in particular Remark 4.27, it becomes apparent that the subgraphs S_w obtained from the FOX algorithm are actually skeletons $S_0(T,x)$ of the associated tree kernel basis vectors. The subgraphs S_w are singular configurations, but x-skeletons in general need not be.

In any case, note the crucial difference: Nut graphs, minimal configurations and singular configurations are concepts for studying the *kernel* of an *arbitrary* graph, whereas the construction of skeletons is limited to *trees* but applicable to *all eigenvalues*. □

Theorem 4.38. [83] Let T be a tree and x an eigenvector for eigenvalue λ of T. Then, $\mathfrak{C}(T - N_\lambda(T,x)) \subseteq \mathfrak{C}(T - N_\lambda(T))$. □

Proof. Let $C \in \mathfrak{C}(T - N_\lambda(T, x))$. Clearly, none of the vertices of C belong to the set $N_\lambda(T)$. Therefore C is a subgraph of some component $C' \in \mathfrak{C}(T - N_\lambda(T))$. According to Lemma 4.35 the vector $x|_{C'}$ is an eigenvector for eigenvalue λ on C' and does not have any zero entries on C'. Hence $C = C'$. ■

Corollary 4.39. [83] Let x, x' be eigenvectors for eigenvalue λ of a given tree and let $C \in \mathfrak{C}(T - N_\lambda(T, x))$, $C' \in \mathfrak{C}(T - N_\lambda(T, x'))$. Then either C and C' are identical or they are disjoint subgraphs of T. □

Corollary 4.40. [83] Let T be a tree with eigenvector x for eigenvalue λ. Then,

$$N_\lambda^C(T, x) \subseteq N_\lambda(T).$$

□

Corollary 4.41. [83] Let T be a tree with eigenvalue λ. Then,

$$\mathfrak{C}(T - N_\lambda(T)) = \bigcup_x \mathfrak{C}(T - N_\lambda(T, x)),$$

where the union is taken over all eigenvectors x for eigenvalue λ of T. □

As a consequence of Corollary 4.41 we can safely merge the x-skeleton forests of an entire eigenspace. Let T be a tree and let $C_1, \ldots, C_r$ be the elements of $\mathfrak{C}(T - N_\lambda(T))$. Let the associated contracted vertices in $T/\{C_i\}_{i=1}^r$ be $c_1, \ldots, c_r$. Denote the union of the sets $N_\lambda^C(T, x)$ by $N_\lambda^C(T)$. Now, we define the *skeleton* $S_\lambda(T)$ as the sub-forest of $T/\{C_i\}_{i=1}^r$ induced by the vertices of $N_\lambda^C(T) \cup \{c_1, \ldots, c_r\}$.

In Figure 4.7 an example of a tree T with threefold eigenvalue 2 is shown along with its skeleton forest $S_2(T)$. The black vertices of T denote the vertices on which the respective eigenvector vanishes. It can be clearly seen how the respective components of $T - N_\lambda(T, x)$ correspond to a part of the skeleton. The black vertices in the skeleton correspond to the set $N_2^C(T)$.

Lemma 4.42. [83] Let T be a tree with eigenvalue λ. Then $\mathfrak{C}(T - N_\lambda^C(T))$ can be partitioned into $\mathfrak{C}(T - N_\lambda(T))$ and a set of trees without eigenvalue λ. □

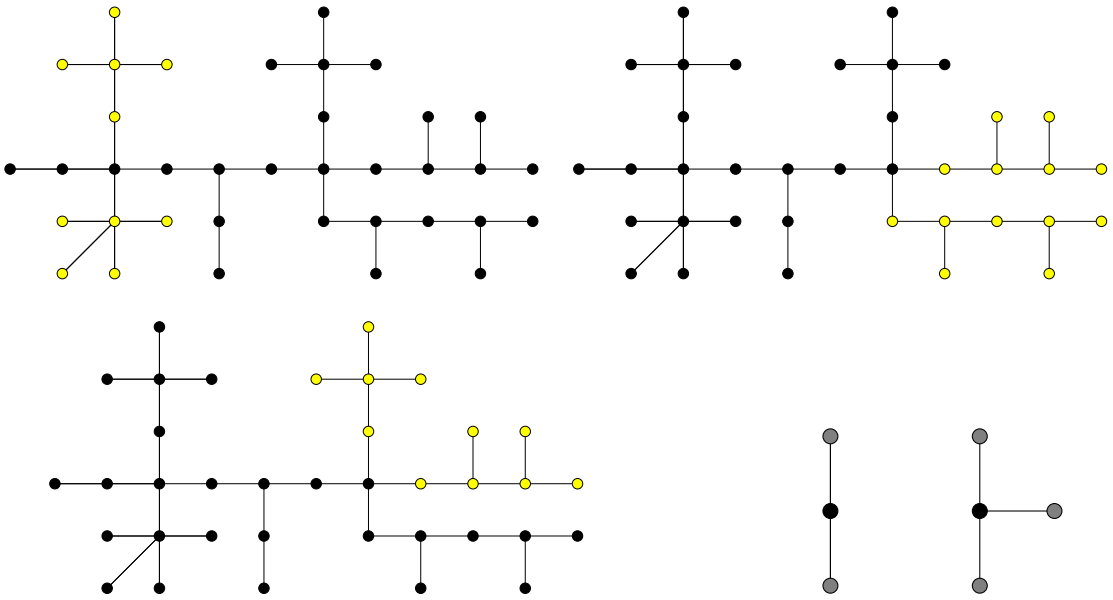


Figure 4.7: Eigenvector zero-nonzero patterns of a tree and corresponding skeleton forest

Proof. Considered as a subgraph of T, every component of $T - N_\lambda(T)$ is adjacent only to vertices from $N_\lambda^C(T)$, but by definition does not contain such vertices. So $\mathfrak{C}(T - N_\lambda(T)) \subseteq \mathfrak{C}(T - N_\lambda^C(T))$. By construction all elements of $\mathfrak{C}(T - N_\lambda(T))$ have eigenvalue λ. Now, let $C \in \mathfrak{C}(T - N_\lambda^C(T)) \setminus \mathfrak{C}(T - N_\lambda(T))$. All vertices of C necessarily belong to the set $N_\lambda(T)$ so that every eigenvector of T for eigenvalue λ must vanish on C.

Assume that there exists an eigenvector y of C for eigenvalue λ. Construct an eigenvector z for eigenvalue λ of T as follows. Firstly, set $z|_C = y$. Consider a vertex w that is adjacent to C in T and let v be the neighbour of w in C. Let ν be the value of y on v.

Case $\nu = 0$: Simply set z to zero on the vertices of the particular component of $T - C$ that contains w.

Case $\nu \neq 0$: Clearly, $w \in N_\lambda^C(T)$ so that by construction, w has a neighbour $u \neq v$ that belongs to a component of $T - N_\lambda(T)$ (since v does not). There exists an eigenvector x of T for eigenvalue λ that vanishes on C and w but does not vanish on u. We may assume c that x has value $-\nu$ on u. Let T_u be the branch of T connected to w via u. Let T_F be the union of the branches connected to w via the neighbours of w different from u, v. Note that T_F is nonempty since x must fulfil the summation rule at vertex w. Set $z|_{T_F} = 0$ and $z|_{T_u} = x|_{T_u}$. Now, the summation rule holds for w and all the vertices of T_u and T_F.

Apply the described procedure for every eligible vertex w. After that the values of z are completely determined. This yields a valid eigenvector for eigenvalue λ of T

that does not vanish on C, a contradiction. ■

Combining Lemma 4.42 with Corollary 4.41 and Lemma 4.35 we can derive the following useful statement:

Lemma 4.43. [83] Let T be a tree and x an eigenvector for eigenvalue λ of T. Then for every $C \in \mathfrak{C}(T - N_\lambda^C(T))$ the restriction $x|_C$ either has only zero entries or only non-zero entries. In the latter case it constitutes an eigenspace basis of the subgraph C of T. □

Observe that every vector from the null space of some x-skeleton can be trivially extended to a vector from the null space of the corresponding skeleton $S_\lambda(T)$.

For the following lemma note that by Lemma 4.42 the skeleton $S_\lambda(T)$ can be considered as a subgraph of the tree $T/\mathfrak{C}(T - N_\lambda^C(T))$.

Lemma 4.44. [83] Let T be a tree with eigenvalue λ. Further let $S' = T/\mathfrak{C}(T - N_\lambda^C(T))$ and $S = S_\lambda(T)$. Then the skeleton S forms an induced sub-forest of the tree S' such that $S' - V(S)$ contains no edges. □

Proof. Assume that S does not form an induced subgraph of S'. Then two vertices of S are adjacent in S' but not already in S. Since these vertices by construction must lie in the same component of S, the additional edge would create a cycle in S', which is impossible. By construction the vertices of $S' - V(S)$ are mutually non-adjacent in S'. ■

Lemma 4.44 allows us to derive the notion of a *meta skeleton* in which the components of the skeleton forest of a tree T with eigenvalue λ are joined by exactly those vertices contracted from the component trees of $\mathfrak{C}(T - N_\lambda^C(T))$ that do not have eigenvalue λ (cf. Lemma 4.42).

Next we explore the relation between eigenspace bases of trees and null space bases of the respective skeleton forests.

Construction 4.45. [83] Let $B = \{b_1, \ldots, b_r\}$ be an eigenspace basis for eigenvalue λ of a given tree T. Construct a basis $B' = \{b'_1, \ldots, b'_r\}$ of the same eigenspace as follows. Let initially $b'_i = b_i$ for $i = 1, \ldots, r$ and let $M = \emptyset$. There exists a component $C_1 \notin M$ of $T - N_\lambda(T)$ such that $b_1|_{C_1}$ does not vanish. By Lemma 4.43 we can subtract suitable multiples of b'_1 from $b'_2, \ldots, b'_r$ such that $b'_i|_{C_1} = 0$ for $i = 2, \ldots, r$. Add C_1 to the set M. Proceed iteratively for b_j, $j = 2, \ldots, r$, by in turn finding a suitable $C_j \notin M$ and establishing $b'_i|_{C_j} = 0$ for $i = j + 1, \ldots, r$. □

The previous construction immediately gives rise to the following observation.

Observation 4.46. [83] Let T be a tree and let λ be an eigenvalue of T with multiplicity $r \geq 1$. Then, $|\mathfrak{C}(T - N_\lambda(T))| \geq r$. □

We say that a set $\{x_1, \ldots, x_r\}$ of eigenvectors for eigenvalue λ of a tree T is *straight* if the components of $T - N_\lambda(T)$ can be numbered $C_1, \ldots, C_s$ such that for $j = 1, \ldots, r$ we have $x_j|_{C_j} \neq 0$ but $x_j|_{C_i} = 0$ for $i = j + 1, \ldots, r$. Observation 4.46 guarantees that $s \geq r$. Note that by Lemma 4.43 each condition $x_j|_{C_j} \neq 0$ actually means that x_j vanishes on none of the vertices of C_j. By Construction 4.45 every tree eigenspace has a straight basis.

Lemma 4.47. [83] Every straight set of tree eigenvectors is linearly independent. □

Proof. Let $X = \{x_1, \ldots, x_r\}$ be a straight set of eigenvectors for eigenvalue λ of a tree T. Let $u_1, \ldots, u_r$ be vertices of T such that they belong to distinct subgraphs of T representing components of $T - N_\lambda(T)$. There exist enough such vertices because of Observation 4.46. Let M be the matrix formed by taking $x_1, \ldots, x_r$ as columns and then retaining only those rows that correspond to the entries of the x_i on the vertices $u_1, \ldots, u_r$. If X is straight then with respect to a suitable numbering of the x_i the vertices u_i can be selected such that M is a lower diagonal matrix with non-zero diagonal entries. Hence, the set X is linearly independent. ■

Theorem 4.48. [83] Let T be a tree with eigenvalue λ and corresponding eigenspace basis B. Then for every vector $b \in B$ there exists a vector b' from the null space of the skeleton $S_\lambda(T)$ such that b' is non-zero exactly on the vertices corresponding to the contracted subgraphs of T on which its associated vector $b \in B$ does not vanish. If B is straight, then the vectors created from B are linearly independent. □

Proof. Let $b \in B$ and initialise $b' = 0$. In the following let $C(v)$ denote the contracted subgraph corresponding to a vertex v of $S_\lambda(T) - N_\lambda^C(T)$. Moreover, if two vertices from $S_\lambda(T) - N_\lambda^C(T)$ have a common neighbour in $S_\lambda(T)$ (necessarily from $N_\lambda^C(T)$) they are called *brothers*.

For every component S of $S_\lambda(T)$ proceed as follows. Fix a vertex s of $S - N_\lambda^C(T)$. If b is non-zero on $C(s)$, then set b' to 1 on s. Consider s as visited and all other vertices of $S - N_\lambda^C(T)$ as unvisited. We now employ a tree search that starts at s and iteratively corrects the values of b' on the vertices of $S - N_\lambda^C(T)$ such that, finally, $b'|_S$ belongs to the null space of S and assumes the desired zero-nonzero pattern. The search only visits unvisited brothers of already visited vertices.

Let v be a visited vertex of $S - N_\lambda^C(T)$ that has unvisited brothers. Mark all brothers of v as visited once the steps described below have been carried out. Let $W \subseteq N_\lambda^C(T)$ contain all vertices that are adjacent (in $S_\lambda(T)$) both to v and some unvisited brother of v. Now, iterate over the vertices $w \in W$. Let $v_1, \ldots, v_r$ be all those unvisited brothers of v that are adjacent to w and for which b does not vanish on $C(v_i)$. By construction, each vertex v_i has exactly one visited brother, namely v. Observe at this point that we have necessarily $r \geq 1$ if b does not vanish on $C(v)$ because else the summation rule would fail for b on the vertex in T that corresponds to w. Hence, it is always possible to assign suitable non-zero values to the vertices $v_1, \ldots, v_r$ such that b' fulfils the summation rule for vertex w.

By construction and the definition of a skeleton it follows immediately that b' is a valid eigenvector from the null space of $S_\lambda(T)$. Its zero-nonzero pattern is as claimed. If B is straight, then the set of vectors created from all the vectors of B using the above procedure is straight as well. Therefore it is linearly independent by Lemma 4.47. ■

Since every tree eigenspace has a straight basis we can immediately relate the dimensions of a tree eigenspace and the null space of the associated skeleton:

Corollary 4.49. [83] Let T be a tree with eigenvalue λ of multiplicity $r \geq 1$. Let s be the nullity of $S_\lambda(T)$. Then $r \leq s$. □

Theorem 4.50. [83] Let T be a tree with eigenvalue λ and let B' be a basis of the null space of its skeleton $S_\lambda(T)$. Then for every vector $b' \in B'$ there exists an eigenvector b of T for eigenvalue λ such that b is non-zero exactly on those subgraphs of T that correspond to vertices of $S_\lambda(T)$ on which b' does not vanish. If B' is straight, then the vectors created from B' are linearly independent. □

Proof. In view of Lemma 4.43 it is possible to use a technique similar to the one used in the proof of Theorem 4.48, just in the opposite direction. ■

Corollary 4.51. [83] Let T be a tree with eigenvalue λ of multiplicity $r \geq 1$. Let s be the nullity of $S_\lambda(T)$. Then $r \geq s$. □

By Corollaries 4.49 and 4.51 we see that the multiplicity of the eigenvalue λ of a tree T equals the nullity of the skeleton $S_\lambda(T)$. As we have seen in section 4.1, the nullity of a forest is closely linked to its matching properties. We will exploit these ties with respect to skeletons. But first let us explicitly relate maximum matchings of trees to eigenvectors of their null spaces:

Theorem 4.52. [83] Let T be a tree with edge set E. Let K contain all vertices of T that may be missed by some maximum matching of T. Further, let N contain all vertices of T that are saturated by all maximum matchings of T. Consider a fixed maximum matching M of T and let $K_M \subseteq K$ be the vertices missed by M.

Then a simply structured null space basis of T can be constructed as follows. Pick a vertex $v \in K_M$ and find the subtree S_v of T formed by the union of all maximal paths that start at v and alternatingly contain edges from $E \setminus M$ and M, such that each edge in the path is incident to one vertex from N and one from $K \setminus (K_M \setminus \{v\})$. Assign weight 1 to all vertices of S_v whose distance to v is divisible by four, assign weight -1 if the distance is two modulo four, and assign zero to all other vertices of T. □

Proof. This is simply a reformulation of Theorem 4.21 in term is matchings, by virtue of Lemma 4.24, Lemma 4.25, and Corollary 4.26. ■

Corollary 4.53. [83] Let T be a tree. Then the set of vertices saturated by all maximum matchings of T is exactly the set of vertices on which every vector from the null space of T vanishes. □

Corollary 4.54. [83] Let T be a tree and let R be the set of those vertices of T on which the null space of T does not completely vanish. Then the nullity of T equals the number of connected components of the subgraph of T induced by the set R minus the number of vertices of T that are adjacent to R but not contained in it. □

We will revisit Corollary 4.54 later on in section 4.5.3.

Theorem 4.55. [83] Let T be a tree with eigenvalue λ.

Then the set of vertices of the skeleton $S_\lambda(T)$ that may be missed by a maximum matching of the skeleton consists exactly of the vertices corresponding to the contracted components of $T - N_\lambda(T)$.

The number of vertices of $S_\lambda(T)$ that are missed by a maximum matching of the skeleton equals the multiplicity of eigenvalue λ of T.

The non-zero entries of a vector from the null space of $S_\lambda(T)$ only occur on vertices that correspond to the contracted elements of $\mathfrak{C}(T - N_\lambda(T))$. □

Proof. This follows from Corollary 4.49, Corollary 4.51, Theorem 4.52 and Corollary 4.53. ■

Remark 4.56. Given a tree, a λ-skeleton and a maximum matching of the skeleton, we can determine the multiplicity of λ as an eigenvalue of T by Theorem 4.55. Moreover, since Theorem 4.52 allows to construct a straight basis of the skeleton null space from a given maximum matching, a basis of the corresponding tree eigenspace can be obtained constructively by means of Theorem 4.50.

The only catch is that usually the skeleton vertices that may or may never be missed by a maximum matching are not known beforehand. Luckily, the FOX algorithm presented in section 4.1 allows to generate maximum matchings along with the sets K,N as required by Theorem 4.52. Every maximum matching can be obtained by a suitable run of the FOX algorithm. □

Now we derive an interesting feature of a skeleton that becomes important once we want to characterise what forests may actually occur as skeletons (as we will see later in the proof of Theorem 4.60):

Lemma 4.57. A skeleton $S_\lambda(T)$ of a tree T with eigenvalue λ does not contain any edges that belong to every maximum matching of the skeleton. □

Proof. Theorem 4.48, Theorem 4.52 and the definition of $S_\lambda(T)$ imply that a pair of skeleton vertices can only be adjacent if one of them is never missed by a maximum matching whereas the other one may be missed by a maximum matching of the skeleton. ■

Concluding this section, it is interesting to note that the skeleton construction cannot be arbitrarily iterated in the sense that a skeleton is its own eigenvalue 0 skeleton:

Lemma 4.58. Let T be a tree with eigenvalue λ and let $S = S_\lambda(T)$ its λ-skeleton. Then the skeleton $S_0(S)$ equals S. □

Proof. Clearly zero is an eigenvalue of the skeleton S. Every vertex of S corresponds to a subgraph of $T - N_\lambda(T)$ for which there exists an eigenvector of T for eigenvalue λ that is non-zero on all vertices of the subgraph. So by Theorem 4.48 for every vertex of S that corresponds to an element of $\mathfrak{C}(T - N_\lambda(T))$ there exists a vector from the null space of S that does not vanish on this vertex.

Consequently, the partition of the components of $S - N_0^C(S)$ according to Lemma 4.42 does not contain any trees without eigenvalue zero so that $N_0^C(S) = N_0(S)$.

On the other hand, a component of $S - N_0^C(S)$ that has a null space eigenvector without zero entries must necessarily contain only a single vertex (cf. Theorem 4.52). So every vertex of S is associated with exactly one vertex in the skeleton $S_0(S)$ and no factual contraction of subgraphs of S happens when forming $S_0(S)$. ■

4.5.2 Tree eigenvector composition

In this section a composition technique is outlined that allows to "blow up" a given potential skeleton to a tree whose skeleton is indeed the initial graph. The bottom line is that we can not only use the maximum matching properties of skeletons to describe eigenvectors of a tree but also conversely construct a tree with predetermined eigenvector properties by generating it from a suitable skeleton.

Following the lines of Lemma 4.44, Theorem 4.55 and Lemma 4.57 we introduce the following definition. We call a tree a *meta skeleton* if its vertex set contains an independent set X such that each subtree $C \in \mathfrak{C}(T - X)$ does not have a perfect matching and only vertices of C that are contained in every maximum matching of C are adjacent to vertices of X in T. Moreover, it is required that no edge of $T - X$ is contained in every maximum matching of $T - X$. The set X is called an admissible *non-eigenvalue set* of the meta skeleton (cf. Lemma 4.42).

Construction 4.59. Let S' be a meta skeleton with non-eigenvalue set X. Choose a number $\lambda \in \mathbb{R}$ and construct a tree T as follows. Substitute each vertex of X with a tree without eigenvalue λ. For every component of $S' - X$ replace each of its vertices that may be missed by a maximum matching of the component with a tree that has eigenvalue λ and a corresponding eigenvector without zero entries. Whenever a vertex of an adjacent pair of vertices is substituted with a graph, a single arbitrary vertex of the substituted graph is chosen to become connected to the other vertex of the pair. □

Theorem 4.60. If Construction 4.59 succeeds for a given triplet (S', X, λ), then the generated tree T has eigenvalue λ. Its multiplicity equals the number of vertices missed in a maximum matching of $S' - X$. Moreover, $S_\lambda(T) = S' - X$ and $S' = T/\mathfrak{C}(T - N_\lambda^C(T))$. □

Proof. We show that the skeleton and meta skeleton are as claimed. Then the rest of the theorem follows from Theorems 4.48, 4.50, 4.52, 4.55 as outlined in Remark 4.56.

By the definition of a meta skeleton each component of $S' - X$ has eigenvalue zero so that we can use Theorem 4.52 to obtain a null space basis for each such component. Let K consist of all vertices of S' that may be missed by maximum matchings of

their respective components of $S' - X$. With the same technique that was used in the proof of Theorem 4.38 we can actually find a vector y from the null space of the forest $\bigcup \mathfrak{C}(S' - X)$ that is nonzero on every vertex of K. Let T' be the subgraph of T that is the blown up counterpart of the subgraph $\bigcup \mathfrak{C}(S' - X)$ of S'. Using a technique similar to the one used in the proof of Theorem 4.48 we can now employ the zero-nonzero pattern of y to construct an eigenvector x' for eigenvalue λ of T' since exactly the vertices of S' on which y is non-zero have been blown up to suitable trees. Use zero entries to trivially extend x' to a vector x on T. Then Corollary 4.53 and the meta skeleton definition imply that x is an eigenvector of T for eigenvalue λ since the summation rule clearly also holds for the vertices of the subtrees blown up from the elements of X.

We will see in a moment that the vector x has been chosen such that it is zero exactly on the vertices $N_\lambda(T)$, i.e. $N_=(T, x)N_\lambda(T)$. This fact substantially eases the determination of the skeleton.

Let N be the set of vertices of $S' - X$ that are covered by every maximum matching of $S' - X$. It follows from the definition of the meta skeleton and Lemma 4.57 that every vertex of N is adjacent to some vertex of $S' - X$ that may be missed by a maximum matching of that graph. Since by Construction 4.59 we can consider N also as a subset of the vertices of T it follows immediately that $N \subseteq N_\lambda^C(T)$.

Now assume that there exists an eigenvector z for eigenvalue λ of T that is non-zero on a subtree T_v of T blown up from a vertex of $v \in X$. Since by construction all outer neighbours of T_v in T belong to N the restriction $z|_{T_v}$ must be a valid eigenvector for eigenvalue λ of T_v. But this is impossible by the choice of T_v. Hence, we have $N = N_\lambda^C(T)$, $N_\lambda(T, x) = N_\lambda(T)$ and $\mathfrak{C}(T - N_\lambda(T, x)) = \mathfrak{C}(T - N_\lambda(T))$.

Moreover, if T_X denotes the sub-forest of T that is the union of all graphs blown up from the vertices of X, then $N_\lambda(T) = V(T_X) \cup N$. So $S' - X$ is indeed the skeleton of T and $S' = T/\mathfrak{C}(T - N_\lambda^C(T))$. ■

Remark 4.61. The skeleton property stated in Lemma 4.57 is decisive for choosing proper forests to be blown up. Otherwise, even though valid eigenvectors can be constructed for the blown up graph, it cannot be guaranteed that a proper eigenspace basis is obtained because the skeleton of the blown up graph may in fact not be the graph we expanded. See Figure 4.8 for a malformed skeleton with one-dimensional null space that can be blown up to the graph with eigenvalue 1 shown in Figure 4.6. However, its eigenvalue 1 has multiplicity 3. □

4.5.3 Tree pattern matrices

Let M be a real $n \times n$ matrix. We define a (directed) graph $\Gamma(M)$ with vertices $v_1, \ldots, v_n$ such that there is an edge from v_i to v_j if and only if M has a non-zero

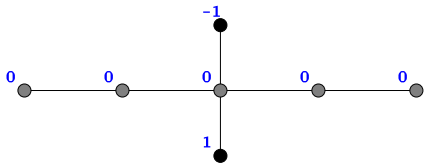


Figure 4.8: Malformed skeleton example

entry at position (i, j). If $\Gamma(M)$ is a tree, then we call M a *tree pattern matrix*. Let $\mathrm{supp}(M; \lambda)$ denote the set of vertices of $\Gamma(M)$ on which the eigenspace for eigenvalue λ of M does not entirely vanish. We call this set the *support* of M with respect to λ. For a graph G, $\mathrm{supp}(G; \lambda)$ denotes the support of its adjacency matrix. Note that it is easy to find examples such that $\mathrm{supp}(M; \lambda)$ and $\mathrm{supp}(\Gamma(M); \lambda)$ are different.

We can extend Corollary 4.54 to the following result which has already been published in [67] but proved differently:

Theorem 4.62. [83] Let M be an $n \times n$ tree pattern matrix. Then the nullity of M equals the number of connected components of the subgraph of $\Gamma(M)$ induced by $\mathrm{supp}(M; 0)$ minus the number of vertices of $\Gamma(M)$ that are adjacent to $\mathrm{supp}(M; 0)$ but do not belong to this set. □

Proof. Let M be a tree pattern matrix and let A be the adjacency matrix of $\Gamma(M)$. Theorem 4.52 states that $\mathrm{supp}(A; 0)$ forms an independent vertex set in $\Gamma(M)$. Given a vector v from the null space of A we can transform it to a vector v' from the null space of M having the same zero-nonzero pattern as follows. Assign v to the vertices of $\Gamma(M)$. Conduct a breadth first search on $\Gamma(M)$ from a fixed vertex s and enforce new summation rules. To be precise, for every vertex z (as traversed by the breadth first search) it is possible to multiply each of its adjacent branches leading away from s with a nonzero factor such that the summation rule given by the line of M that corresponds to z holds. From a straight basis of the null space of A we can thus obtain a straight linearly independent set of vertices from the null space of M. A similar conversion can be employed for the opposite direction. Therefore, $\mathrm{supp}(M; 0) = \mathrm{supp}(A; 0) = \mathrm{supp}(\Gamma(M); 0)$. Now the result follows by Corollary 4.54. ■

In fact, the results from the previous sections allow us to generalise even further. We quoted Lemma 4.34 only as a special case of what is actually proved in [33]. It has been shown that every eigenvector of a tree pattern matrix necessarily belongs to an eigenvalue with multiplicity one if it has no zero entries. Moreover, for the application of the summation rule none of the proofs given in Section 4.5.1 explicitly relied on the fact that it was induced by the adjacency matrix of the tree. Every row of a tree pattern matrix M induces a particular summation rule for the associated

vertex of $\Gamma(M)$. The only difference to the summation rule used for the adjacency matrix is that for every vertex certain non-zero factors are applied to the weights of the neighbours before adding them up. Consequently, we can generalise the entire theory presented in Section 4.5.1 to cover eigenvectors of tree pattern matrices. In particular we obtain the following generalisation of Theorem 4.62:

Theorem 4.63. [83] Let M be an $n \times n$ tree pattern matrix with eigenvalue λ. Then the dimension of the eigenspace of M for eigenvalue λ equals the number of connected components of the subgraph of $\Gamma(M)$ induced by $\mathrm{supp}(M;\lambda)$ minus the number of vertices of $\Gamma(M)$ that are adjacent to $\mathrm{supp}(M;\lambda)$ but do not belong to this set. □

One other noteworthy generalisation is that eigenspace dimensions of tree patterned matrices are determined by sizes of maximum matchings of the respective associated skeletons.

4.6 Eigenspace bases for eigenvalues 1 and −1

It has already been shown in section 4.2 that every tree has a simply structured basis for eigenvalue 0. We complete the characterisation by investigating the other two possible eigenvalues 1 and −1. To this purpose we make use of the concept of decomposing trees by the zero entries of their eigenvectors that was presented in section 4.5. Since trees are bipartite it suffices to restrict further investigations to the eigenvalue 1. Given an eigenspace basis for eigenvalue 1, an eigenspace basis for eigenvalue −1 is readily obtained by negating the signs of all vector entries corresponding to the vertices of one part of a corresponding vertex bipartition.

Examples for eigenvectors for eigenvalue 1 that cannot be scaled to $\{0, 1, -1\}$ entries can be found quite easily — see Figure 4.9, where the claim follows by Lemma 4.34. We will therefore attempt to characterise those trees that have a simply structured eigenspace basis for eigenvalue 1. A simple example of a tree with this property is the path P_5.

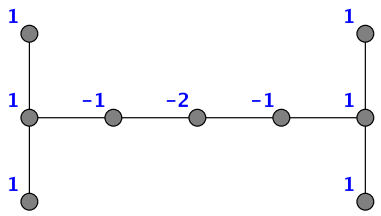


Figure 4.9: Graph without $\{0, 1, -1\}$ eigenvector for eigenvalue 1

Assume that a tree with a simply structured eigenspace basis for eigenvalue 1 is decomposed according to the always-zero entries. Clearly, each such generated com-

ponent has a single eigenvalue 1 and a corresponding eigenvector without zero entries, namely a $\{1, -1\}$ vector. Since such eigenvectors are the building blocks for the composition of trees with simply structured bases for eigenvalue 1 we now direct our attention to them. It turns out that trees with $\{1, -1\}$ eigenvector for single eigenvalue 1 can be characterised in a very elegant way.

4.6.1 $\{1, -1\}$ eigenvectors for eigenvalue 1

Observation 4.64. [83] Let x be an eigenvector for eigenvalue 1 of a given tree T. Then the value of x on a leaf equals the value on its unique neighbour. □

Theorem 4.65. [83] A tree has a $\{1, -1\}$ eigenvector for eigenvalue 1 if and only if the tree can be reduced to a K_2 graph by repeatedly selecting a subgraph as in Figure 4.10 (where the vertices u_0, u_1, w must be leaves in the current reduced graph) and removing all its vertices except z from the current reduced graph. □

Proof. Let T be a tree with $\{1, -1\}$ eigenvector x for eigenvalue 1. Clearly, T must have at least two vertices. If T is a complete graph K_2 there is nothing to show. So we may assume that T has at least three vertices.

Recall that the eccentricity of a vertex is its distance from the graph centre and that the centre of a tree consists of either a single vertex or a pair of adjacent vertices. Let u_0 be a leaf of T that has maximum eccentricity and v its only neighbour. Among those neighbours of v different from u_0 let z be that vertex which is closest to the centre of T. Let $u_1, \ldots, u_r$ be the other neighbours of v besides u_0 and z. Since u_0 has maximum eccentricity the vertices $u_1, \ldots, u_r$ must also be leaves of T.

We may assume that v is not the sole centre vertex of T. Otherwise T would be a star graph $K_{1,r+2}$, which does not have eigenvalue 1. Let w.l.o.g. x have value 1 on u_0. Then by Observation 4.64, x assumes the same value also on the vertices $u_1, \ldots, u_r, v$. The summation rule for vertex v requires a negative value of x on z. Therefore, $r = 1$ and x has value -1 on z.

We now claim that z is adjacent to a leaf with value -1. By the summation rule there exist at least two neighbours of z on which x assumes the value -1. Among these neighbours there exists at least one vertex w such that the branch adjacent to z via the edge wz does not contain any centre vertices of T. Assume that w is not a leaf of T. Then by the summation rule w would have at least one neighbour w' with value 1. Again by the summation rule w' would have at least one neighbour w'' with value 1. But by our assumption about the location of the centre of T the eccentricity of w'' is clearly greater than that of u_0, a contradiction.

Remove the vertices u_0, u_1, v, w from T. Clearly, T remains a tree. Moreover, the summation rule remains valid for all remaining vertices, in particular for z. Since z

has at least one remaining neighbour it follows that T has at least two vertices. We can therefore iterate the reduction step until a graph K_2 has been obtained. The reduction procedure can also be applied for every subgraph of T isomorphic to the one in Figure 4.10 if only u_0, u_1, w are leaves. The maximum eccentricity criterion only asserts the existence of a subtree suitable for reduction.

Conversely, assume that a tree can be decomposed in the described manner. Then we can assemble it from K_2 by iteratively selecting a vertex z and adding vertices u_0, u_1, z, w according to Figure 4.10. The all ones vector forms an eigenspace basis for eigenvalue 1 of the graph K_2. After the addition of the vertices u_0, u_1, z, w we can uniquely augment the previous eigenvector to become a $\{1, -1\}$ eigenvector for eigenvalue 1 of the extended graph. The values on the newly added vertices depend only on the existing eigenvector value on z, cf. Figure 4.10. Iterating this argument we find that T has a $\{1, -1\}$ eigenvector for eigenvalue 1. ■

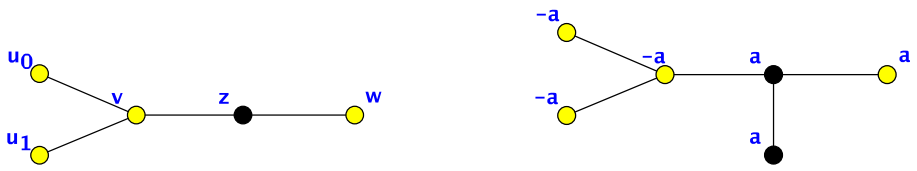


Figure 4.10: Reduction subgraph and weights for $\{1, -1\}$ eigenvectors

Corollary 4.66. [83] There exists a tree with n vertices that has a $\{1, -1\}$ eigenvector for eigenvalue 1 if and only if $n \equiv 2 \mod 4$. □

In the following, let $\mathcal{C}$ denote the class of all trees with $\{1, -1\}$ eigenvector for eigenvalue 1. Note that if a tree has a $\{1, -1\}$ eigenvector for eigenvalue 1, then by Lemma 4.34, the eigenvalue 1 has necessarily multiplicity one.

4.6.2 $\{0, 1, -1\}$ eigenvectors for eigenvalue 1

Having investigated trees with $\{1, -1\}$ eigenvectors it is now straightforward to achieve a characterisation of trees with simply structured eigenspace bases for eigenvalue 1:

Theorem 4.67. [83] Let T be a tree with eigenvalue 1. Then there exists a simply structured basis for the corresponding eigenspace if and only if $C \in \mathcal{C}$ for every component $C \in \mathfrak{C}(T - N_1(T))$. □

Proof. Necessity follows from Lemma 4.43. For sufficiency refer to Theorem 4.65 and consider the reconstruction of (linearly independent) eigenvectors of T from the zero-nonzero patterns of a straight null space basis of its skeleton forest. Simply assign valid $\{1, -1\}$ eigenvectors to all contracted subgraphs of T where the chosen skeleton null space eigenvector is nonzero on the corresponding skeleton vertices. A valid eigenvector is obtained by establishing the summation rule for all vertices of $N_1^C(T)$. This can be achieved by conducting a breadth first search from a fixed nonzero skeleton vertex v. Each time a vertex of $N_1^C(T)$ is visited the summation rule for its partner vertex in T is enforced by suitably multiplying the values on the branches leading away from v. Since the branches have only values from the set $\{0, 1, -1\}$ the only factors that are needed are 1 and -1 so that finally a $\{0, 1, -1\}$ eigenvector is created. ■

In theory, Theorem 4.67 provides us with a completely structural characterisation of all trees whose eigenspace for eigenvalue 1 admits a simply structured basis. The class $\mathcal{C}$ is characterised by a reduction property and the set $N_1(T)$ is independent of the choice of a particular eigenspace basis so that essentially it is an intrinsic structural property of a tree as well.

From a practical point of view, however, there is always an algebraic aspect. In order to check if a tree T has a simply structured eigenspace one would start by computing an arbitrary eigenspace basis for eigenvalue 1 and then try to reduce the components of $T - N_1(T)$. Conversely, the trees with simply structured eigenspace bases for eigenvalue 1 can be generated using Construction 4.59 by using only graphs from $\mathcal{C}$ for the blown-up trees with eigenvalue 1. But for blowing up the vertices of the non-eigenvalue set, trees without eigenvalue 1 are used. So far a non-algebraic characterisation of such trees is unknown. It is even doubtful if such a characterisation exists since it is not difficult to show that every given tree can be extended to a tree with single eigenvalue 1 and a corresponding eigenvector without zero entries. So it seems hard to tell the difference between trees that have eigenvalue 1 and trees that have not. All in all, the desired characterisation of trees with simply structured eigenspace bases has been achieved.

We conclude this section with a construction that allows to derive a simply structured tree eigenspace basis from a given initial eigenspace basis.

Construction 4.68. Let T be a tree with eigenvalue 1 and B a corresponding eigenspace basis. Then a simply structured eigenspace basis for this eigenvalue of T can be obtained as follows:

1. Use B to determine $\mathfrak{C}(T - N_1(T))$ and the skeleton $S_\lambda(T)$.
2. Reduce every component of $T - N_1(T)$ according to Theorem 4.65 and simultaneously determine $\{1, -1\}$ component eigenvectors.
3. Determine a maximum matching of $S_\lambda(T)$, e.g. using one of the algorithms presented in [9], [35] or [78].

4. Construct a straight null space basis B' of $S_\lambda(T)$.

5. Map the vectors of B' to a set of vectors on T by matching their zero-nonzero patterns; to the subgraphs corresponding to nonzero skeleton vertices the respective already computed $\{1, -1\}$ component eigenvectors are assigned, all other vertices are assigned zero values.

6. For every constructed vector use a breadth first search approach to correct the signs of branches such that for every zero value vertex adjacent to a non-zero vertex the summation rule holds.

□

The initial basis B can be obtained by traditional Gaussian elimination, but there exists an algorithm by John and Schild that allows to compute the vectors of B on the tree T itself [55], cf. section 4.4.

5 Unicyclic graphs

Having analysed the eigenspaces of trees we will now consider graphs that contain exactly one cycle, the so-called *unicyclic graphs*.

Applying Theorem 3.1 to the trees emanating from the cycle of a given unicyclic graph, it becomes immediately apparent that a $\{-1, 0, 1\}$ valued eigenvector is only possible if one of the following conditions is satisfied:

1. The corresponding eigenvalue belongs to the set $\{-1, 0, 1\}$.
2. Non-zero values of the eigenvector only occur for vertices of the cycle.

The cases are not exclusive. But if the second condition is satisfied for some eigenvalue $\lambda \notin \{-1, 0, 1\}$, then application of the summation rule reveals that, necessarily, $\lambda \in \{-2, 2\}$. In this case the graph must even be a cycle since the considered eigenvector would necessarily have to be non-zero on *every* vertex of the cycle in the given graph.

For each of the eigenvalues $-2, -1, 0, 1, 2$ one can easily find unicyclic example graphs admitting simply structured eigenspace bases. In the following, we concentrate on eigenvalue 0 and, hence, on the kernel of unicyclic graphs. The introductory remarks and the following results on the kernel indicate that, with due diligence, similar results can be obtained for the other candidate eigenvalues as well.

We have seen in section 4.2 that every graph without cycles admits a simply structured kernel basis. For graphs that contain cycles it easy to find counterexamples — even from the class of unicyclic graphs — that do not admit a simply structured kernel basis, cf. Figure 5.1. It shows an example graph and a basis of its one dimensional kernel.

On the other hand we can easily identify cases when unicyclic graphs admit simply structured kernel bases. According to [7], the adjacency matrix of a unicyclic graph is totally unimodular if and only if the length of its cycle is a multiple of four. Thus, by Theorem 3.4, a simple check on the length of the cycle of a unicyclic graph provides a sufficient criterion for the existence of a simply structured kernel basis. We do not, however, gain any insight regarding basis structure.

The goal of this chapter is to achieve a complete characterisation of all unicyclic graphs admitting simply structured kernel bases. Just as for trees, we rely on the FOX algorithm. The main cases to distinguish result from how the FOX algorithm actually behaves on the vertices of the cycle of the unicyclic graph.

Consider an induced cycle C of a given graph G. Perform the FOX algorithm on G to get $\tilde{D}$ and transfer the node labels to the corresponding nodes of G. We say that C has been *cracked* by this run of FOX if not all of its nodes have been `F` labelled.

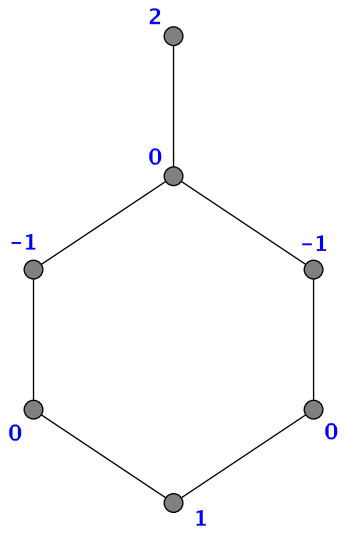


Figure 5.1: Graph without a simply structured kernel basis

Observation 5.1. [82] Let G be a unicyclic graph with cycle C and suppose that C gets cracked by a particular FOX run $\mathcal{A}$. Then, clearly, the first node v of C to become $\overline{\mathtt{F}}$ labelled by run $\mathcal{A}$ must get $\mathtt{0}$ labelled from outside C, i.e. there exists an $\mathtt{X}$ labelled node w in a tree T emanating from C such that v is the outgoing neighbour of w. □

Lemma 5.2. [82] Let G be a unicyclic graph with cycle C. A given FOX run on G does not crack C if and only if all neighbours of C in G have been $\mathtt{0}$ labelled. □

Proof. Assume that a node v from C has a neighbour u in $G - C$ that is either $\mathtt{X}\overline{\mathtt{0}}$ or $\mathtt{F}$ labelled. Since neither of them is $\mathtt{0}$ labelled, u and v must be mutually adjacent in $\tilde{D}$.

Now assume that u is $\mathtt{X}\overline{\mathtt{0}}$ labelled. According to Lemma 4.3 the node v must be its only outgoing neighbour so that by Algorithm 4.1 the node v is $\mathtt{0}$ labelled, a contradiction.

Let therefore u be $\mathtt{F}$ labelled. Let T be the tree in G that is attached to C by the edge uv. Not all neighbours of u within T can be $\overline{\mathtt{X}}\mathtt{0}$ labelled since then u would have exactly one outgoing neighbour and would have become $\mathtt{X}$ labelled by FOX. So we may assume that u has a neighbour in T that is either $\mathtt{X}\overline{\mathtt{0}}$ or $\mathtt{F}$ labelled. As above we see that the case $\mathtt{X}\overline{\mathtt{0}}$ is impossible.

Continuing this argument, we may conclude that there exists a chain of $\mathtt{F}$ labelled nodes starting from v and leading to a leaf of T. But by Lemma 4.3 in $\tilde{D}$ this leaf is

mutually adjacent to its neighbour and therefore has only one outgoing neighbour in $\tilde{D}$. Hence, it would have been `X` labelled by FOX, a contradiction.

The converse statement follows from Observation 5.1. ■

Lemma 5.3. [82] Let G be a unicyclic graph with cycle C. Suppose that C has been cracked by a particular run of FOX. Then C gets cracked by every possible run of FOX. □

Proof. Suppose that there exists a FOX run $\mathcal{A}$ that cracks C and that C has not been cracked by another FOX run $\mathcal{B}$. Determine nodes v, w and a tree T according to Observation 5.1. Rearrange both runs $\mathcal{A}$ and $\mathcal{B}$ in such a way that they start on the nodes of the tree $T+v$. This does not change the resulting digraphs whose node labels we need to consider.

Now focus only on the graph $T+v$. The starting operations of run $\mathcal{A}$ represent a partial FOX run on $T+v$ which may be completed in a way such that v, w becomes an `XO` labelled pair. On the other hand, the operations of run $\mathcal{B}$ represent a complete FOX run on $T+v$ that creates an `F` label for v. Hence, we have a contradiction to Lemma 4.19. ■

As a consequence of the previous lemma, we say that the cycle C of a unicyclic graph G is either cracked or uncracked.

The following investigations will treat these cases separately. To begin with, we consider uncracked unicyclic graphs.

5.1 Uncracked cycles

Lemma 5.4. [82] Let G be a unicyclic graph with uncracked cycle C. If $x \in \ker G$, then $x_C \in \ker C$ holds for the restriction x_C of x to C. □

Proof. According to Lemma 5.2 the neighbours of C in $G-C$ are all `O` labelled. Therefore, the respective node weights must be zero for every vector from the kernel of G. Consequently, we see that on the nodes of C the eigenvector summation rules for C and G coincide. ■

Theorem 5.5. [82] Let G be a unicyclic graph with uncracked cycle C. If the size of C is not a multiple of four, then G has a simply structured kernel basis. □

Proof. Since the size n of C is not divisible by four, the cycle graph C_n is nonsingular [10]. Therefore, according to Lemma 5.4 every vector from $\ker G$ must vanish on the vertices of C. Consider the forest $G - C$ and let T be one of its trees. Then we see by Lemma 5.2 that $x_T \in \ker T$ must hold for the restriction x_T of any vector $x \in \ker G$ to T. Hence, a simply structured kernel basis of G is obtained by means of trivial embedding after using Construction 4.16 to determine simply structured kernel bases for the trees of $G - C$. ■

Lemma 5.6. [82] Let n be a multiple of four and let $x \in \ker C_n$. Then x takes values $c, 0, -c, 0, c, 0, -c, \ldots$ for some $c \in \mathbb{R}$ on consecutive vertices of C_n. □

Proof. This is a well-known result that follows directly from the summation rule. ■

Theorem 5.7. [82] Let G be a unicyclic graph with uncracked cycle C. If the size of C is a multiple of four, then G has a simply structured kernel basis. Moreover,

$$\dim \ker G = \dim \ker(G - C) + 2.$$

□

Proof. Since the size n of C is divisible by four, the cycle graph C_n is singular. Therefore, according to Lemma 5.4 every vector from $\ker G$ takes values $c, 0, -c, 0, c, 0, -c, \ldots$ for some $c \in \mathbb{R}$ on the consecutive vertices of C. If $c = 0$, then we may proceed as in the proof of Theorem 5.5. For $c \neq 0$ we deduct from Lemma 5.6 that there are only two linearly independent choices of possible weights on C (the zero-nonzero pattern is rotated by one position). We may assume that these weights are only from the set $\{0, 1, -1\}$. Given such weights on C we need to construct a valid vector from $\ker G$. Assign zero weights to the vertices of all trees emanating from the zero weight vertices of C. Because of Lemma 5.2 we may employ Lemma 4.28 to construct a suitable eigenvector for each tree emanating from a vertex with weight ± 1.

This construction yields $\dim \ker(G-C)+2$ linearly independent vectors from $\ker G$. These vector even form a basis of $\ker G$. To see this, it suffices to note that B cannot contain two vectors that do not vanish on C but have the same zero-nonzero pattern on C because otherwise a suitable linear combination would yield a vector extended from $\ker(G - C)$. ■

An example that illustrates Theorem 5.7 can be found in Figure 5.2.

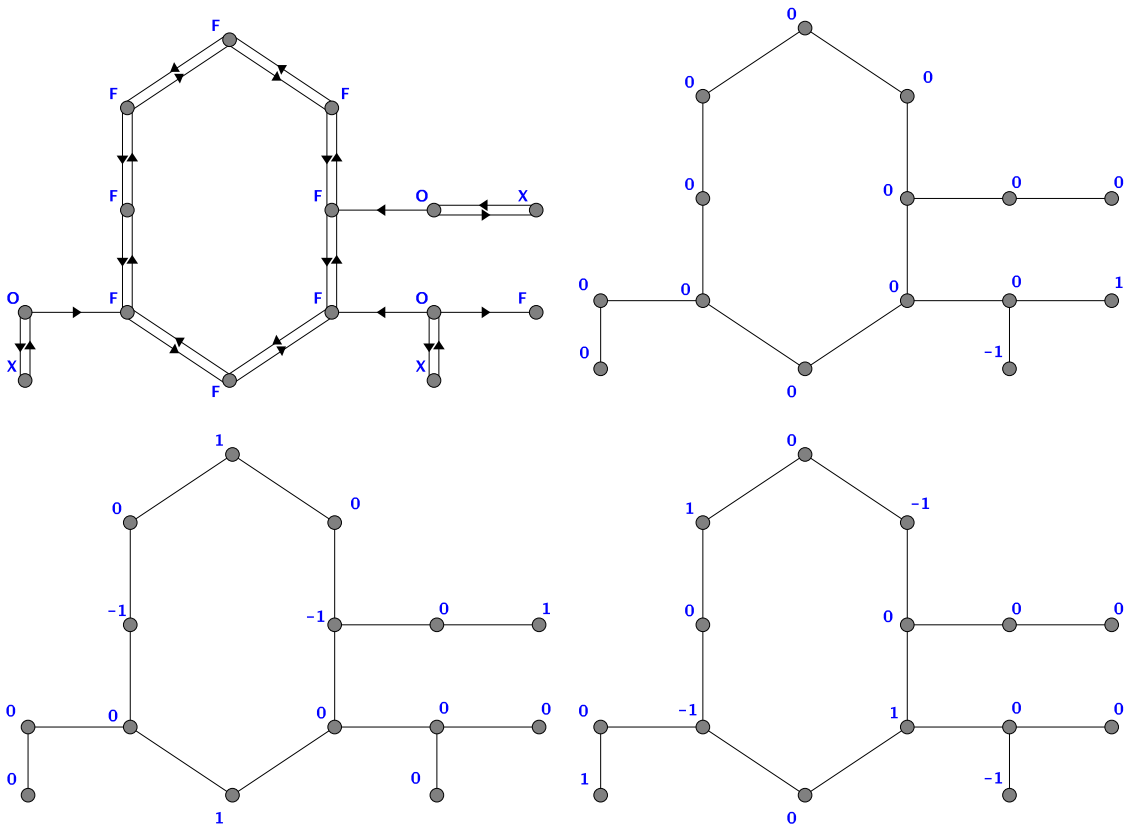


Figure 5.2: Kernel basis of a unicyclic graph according to Theorem 5.7

5.2 Cracked cycles

Let us consider the case of unicyclic graphs with cracked cycle. It turns out that it demands more effort than the uncracked case.

Our next theorem determines the size of the kernel of a unicyclic graph with cracked cycle in terms of the number of `F` labelled nodes. Thus, the result relates to Theorem 4.21.

But beforehand, we require the definition of a trivial embedding function ι. Let H be a subgraph of G and let $w : H \to \mathbb{R}$ be a node weight function on H. Then, $\iota(w) : G \to \mathbb{R}$ is defined by

$$\iota(w)(x) = \begin{cases} w(x) & \text{if } x \in V(H) \\ 0 & \text{else} \end{cases}.$$

Naturally, we assume that ι works accordingly for weight vectors (with respect to a fixed vertex numbering).

Theorem 5.8. [82] Let G be a unicyclic graph with cracked cycle C. Let therefore

u be an $\mathtt{O}$ labelled node of C whose $\mathtt{X}$ labelled partner v does not lie on C. Then,

$$\ker G \subseteq \iota(\ker(G-u))$$

and

$$|H_{\mathtt{F}}| = \dim\ker G = \dim\ker(G-u) - 1.$$

□

Proof. There exists a FOX run on G such that u is the first node of C that receives a label. Consider the situation just after u has been $\mathtt{O}$ labelled. Since the selection of a node to be $\mathtt{X}$ labelled does not depend on its incoming edges it is clear that the outgoing edges of u leading to still unlabelled nodes have no effect on the remaining steps of the FOX run considered. We may therefore remove the corresponding edges from G and still get the same node labels. If we further disconnect u and v we see that v has no more incoming neighbours and must therefore become $\mathtt{F}$ labelled. Consequently, there exists a FOX run on $G-u$ that creates the same node labels as on G before, only that v is $\mathtt{F}$ labelled. Since $G-u$ is a forest it follows from Theorem 4.21 that the number of $\mathtt{F}$ labelled nodes of G exceeds the dimension of $\ker(G-u)$ by exactly one.

Since u is $\mathtt{O}$ labelled it holds necessarily that $x|_{G-u} \in \ker(G-u)$ for every vector $x \in \ker G$. It follows with Lemma 4.10 that $\ker G$ consists of exactly those vectors $x \in \iota(\ker(G-u))$ that obey the summation rule for the node u. The summation rule poses an at most one-dimensional restriction on $\iota(\ker(G-u))$. If the summation rule did not pose a restriction, then every vector from $\ker G$ would have to be zero on the neighbours of u, in particular on v. But according to Theorem 4.21 there exists a vector from $\iota(\ker(G-u))$ that does not vanish on v, a contradiction. ■

Note that Theorem 5.8 does not yet guarantee the existence of simple kernel bases since the component eigenvectors may have to be multiplied to suit the summation rule at the node u.

At this point we can provide a generalisation of Lemma 4.19:

Lemma 5.9. [82] Let G be a unicyclic graph. Then the set $V(H_{\mathtt{O}})$ is invariant for all possible runs of FOX on G and contains exactly those vertices on which every vector from $\ker G$ vanishes. □

Proof. From the proofs of Theorems 5.5, 5.7 and 5.8 it follows that for every $\overline{\mathtt{O}}$ labelled node (with respect to a given FOX run) there exists a vector from the kernel of G that does not vanish on this node. Recalling Lemma 4.10, the result follows. ■

Continuing our analysis of the cracked case we will now investigate node separation issues since these lead to further sub-cases we need to consider.

Lemma 5.10. [82] Let G be a unicyclic graph with cracked cycle C. Then the following statements are equivalent for a given FOX run:

1. No two adjacent nodes of C are assigned `O` labels.
2. Adjacent nodes of C do not become separated in $\tilde{D}$.

□

Proof. Nodes on C that become separated in $\tilde{D}$ must necessarily be `O` labelled since their incoming edges get deleted.

Conversely, assume that there exists a pair x, y of adjacent `O` labelled nodes on C. We will show that there exist two nonadjacent nodes in $\tilde{D}$ whose counterparts in C are adjacent.

Case 1. Suppose that both x and y are `X` labelled as well. Let y' be the second neighbour of x on the cycle C. Then according to Lemma 4.3 the node x and either y or y' belong to different weak components of $\tilde{D}$.

Case 2. Suppose that at most one of the nodes x and y is `X` labelled. Then by Lemma 4.2 the nodes x and y belong to different weak components of $\tilde{D}$. ■

The previous lemma motivates the following definition. We say that a cycle C of a graph G is cut by a given FOX run if there exist two nodes of C that are adjacent in G but disconnected in $\tilde{D}$.

It follows directly from Lemma 5.9 that cutting the cycle does not depend on the particular FOX run so that it makes sense to say that a unicyclic graph either has a cut or uncut cycle. A cut cycle is necessarily cracked.

Note that if an even cycle gets cut, then separation occurs for at least two pairs of neighbours since between the separated nodes there must be alternating `O` and $\overline{\texttt{O}}$ labels along the nodes of the cycle.

Corollary 5.11. [82] Let G be a unicyclic graph with cut cycle C. Then G has a simply structured kernel basis. □

Proof. By Lemma 5.10 we see that D is a forest so that the result follows from Lemma 4.9 and Theorem 4.21. ■

Corollary 5.12. [82] Let G be a unicyclic graph with uncut cracked cycle C. Then C is even. □

Proof. According to Lemma 5.10 the nodes of C do not become separated in $\tilde{D}$. Thus, O labelled and $\overline{\text{O}}$ labelled nodes must alternate on C so that C must necessarily be even. ■

Corollary 5.13. [82] Let G be a unicyclic graph with cracked odd cycle C. Then G has a simply structured kernel basis. □

Let us take a closer look at unicyclic graphs with uncut cracked cycles. Before we analyse their kernel structure let us determine when a cracked cycle remains uncut.

We say that the cycle C of a unicyclic graph G can be cracked at node u by node v if u lies on C, v lies outside C and there exists a FOX run on G such that v is the X labelled partner v of u.

Lemma 5.14. [82] Let G be a unicyclic graph with cracked cycle C. Then C remains uncut if and only if C is even and each pair of nodes at which C may be cracked has even distance in G. □

Proof. Sufficiency follows directly from the proof of Corollary 5.12 and Lemma 5.9.

Conversely, construct a FOX run on G that avoids the creation of X labelled nodes on C for as long as possible. It is obvious that when the first node of C becomes X labelled there are no more unlabelled nodes on the tree emanating from C so that C has been cracked at all possible nodes. Note that between any two O labelled nodes of C there lies an even number of unlabelled nodes.

Subsequently, let the FOX run avoid labelling nodes outside C for as long as possible by repeating the following procedure. On C select an arbitrary unlabelled neighbour v of an O labelled node. If v has no outgoing neighbour, pick again. Otherwise v has exactly one outgoing neighbour, which by construction must be its second neighbour y on C. To see this it suffices to note that since C cannot be cracked at v its neighbours outside C must be O labelled. Consequently, let FOX select v for X labelling. Thus we have reduced the gap of unlabelled nodes between two particular O labelled nodes on C by two.

It is clear that this procedure leads to a strict O-$\overline{\text{O}}$ pattern on C that does not cut C. Employing an argument similar to that in Lemma 4.8 it is guaranteed that the final steps of the FOX run may introduce multiple labels but does not further change the set of O labelled nodes. ■

Theorem 5.15. [82] Let G be a unicyclic graph with uncut cracked cycle C. If the size of C is a multiple of four, then G has a simply structured kernel basis. Moreover, this basis may be chosen to contain at most one vector x with $x_C \in \ker C \setminus \{0\}$. □

Proof. Let u be an `O` labelled node of C whose `X` labelled partner v does not lie on C. Since $G - u$ is a forest, we may use Construction 4.16 to obtain a simply structured kernel basis B of $G-u$. We assume that the FOX run on $G-u$ needed for the construction has been derived from a FOX run on G as described in the proof of Theorem 5.8, yielding the same node labels except for v. Let $B = \{b_0, \ldots, b_k\}$ with $k = |H_{\mathtt{F}}|$, cf. Theorem 5.8. By construction, the restriction of B to $H_{\mathtt{F}}$ yields the standard unit basis. Assume w.l.o.g. that b_0 has value 1 on v. Since G is unicyclic it follows that one component of $G - u$ contains exactly two neighbours of u in G whereas all other components contain at most one neighbour. Therefore, every vector b_i may have at most two nonzero weights on the neighbours of u in G.

We will now construct a simply structured linearly independent subset B' of $\ker G$ with $B' = \{b'_1, \ldots, b'_k\}$. Theorem 5.8 then ensures that B' is a basis of $\ker G$. Consider each vector $b_1, \ldots, b_k$ separately and let n_i be the number of nonzero weights among the neighbours of u in G for the vector b_i.

Case $n_i = 1$. Let y_1 be the only neighbour of u with nonzero weight (assume w.l.o.g. that this weight equals one). Then $\iota(b_i - b_0) \in \ker G$. Let therefore $b'_i = \iota(b_i - b_0)$.

Case $n_i = 2$. Let y_1, y_2 be the neighbours of u with nonzero weight. Because of Construction 4.16 both y_1 and y_2 lie in the same subgraph S_w of $G - u$ for some `F` labelled node w.

But since the `X` labelled partner v of u does not lie in C it follows that there exists no directed path from y_1 to y_2 via u in $\tilde{D}$ and vice versa. So $C - u$ is necessarily a subgraph of S_w because otherwise the propagation of weights starting from w could not have succeeded. Since the size of C is a multiple of four, we deduce from Construction 4.16 that $\mathrm{dist}_{G-u}(y_1, y_2) \equiv 2 \mod 4$ so that b_i assigns opposite weights to y_1 and y_2. This implies $\iota(b_i) \in \ker G$. Let therefore $b'_i = \iota(b_i)$.

Now suppose that B' contains two nonzero vectors b_{i_1} and b_{i_2} whose restriction to C lies in $\ker C$. It follows from Lemma 5.6 and Lemma 4.10 that there exists a linear combination b' of b_{i_1} and b_{i_2} that vanishes on C. Substitute b' for b_{i_2} in B'. Clearly, B' retains its basis property. Repeat the procedure until B' only contains at most one vector x with $x_C \in \ker C \setminus \{0\}$. ■

We have seen in the proof of Theorem 5.15 that there may exist an `F` labelled node w such that for some `O` labelled node u from C its neighbours y_1 and y_2 in C both lie in S_w. We may ask what happens if there exist several such nodes w, cf. Figure 5.3.

Lemma 5.16. [82] Let G be a unicyclic graph with cracked cycle C. Let u be an `O` labelled node of C whose `X` labelled partner v does not lie on C and let y_1 and y_2 be the neighbours of u in C.

If there exist distinct nodes w_1 and w_2 such that y_1 and y_2 both lie in S_{w_1} and S_{w_2}, then w_1 and w_2 lie in the same tree emanating from C. □

Proof. According to the definition of S_{w_1} there exist directed paths $\tilde{P}_1$ and $\tilde{P}_2$ in $\tilde{D}$ leading from y_1 and y_2 to w_1. Clearly, $\tilde{P}_i$ cannot contain u. If w_1 was situated on the cycle C it would be impossible to reach any other `F` labelled node from both y_1 and y_2. Therefore, w_1 lies outside C so that necessarily $C - u$ is a subgraph of S_{w_1}. The paths $\tilde{P}_1$ and $\tilde{P}_2$ start separately but then unite in a common final node z on C before they leave the cycle to enter the emanating tree T that contains w_1. Note that necessarily z is $\mathtt{X}\overline{\mathtt{O}}$ labelled. In order to reach w_1 from z it is required that the `O` labelled partner of z lies in T. This leaves z with no outgoing nodes on C, rendering it the only node on C that can be reached both from y_1 and y_2. Thus, any other `F` labelled node that can be reached from both y_1 and y_2 must lie in T as well. ■

Corollary 5.17. [82] Let G be a unicyclic graph with cracked cycle C. Let u be an `O` labelled node of C whose `X` labelled partner v does not lie on C and let y_1 and y_2 be the neighbours of u on C.

If there exists an `F` labelled node w such that both y_1 and y_2 lie in S_w, then C is even. □

Proof. This follows from the proof of Lemma 5.16 and Lemma 4.11. ■

Let us carry our analysis of the structures S_w a little further.

Lemma 5.18. [82] Let G be a unicyclic graph with cracked cycle C. Let u,u' be two nodes at which C may be cracked. Then there exists a FOX run on G such that the neighbours y_1, y_2 of u within C do not lie in a common subgraph S_w for any `F` labelled node w. □

Proof. Since C may be cracked at u and u' it is possible to construct a FOX run on G such that u and u' are the first nodes on C to be labelled, namely `O` labelled. Now consider adjacencies of u and u' in $\tilde{D}$. Any other node of C may at best be an outgoing neighbour of u or u', but never an incoming neighbour. Therefore there cannot exist directed paths in $\tilde{D}$ from y_1 and y_2 leading to the same `F` labelled node w since at least one of these paths would have to run via u or u', which is impossible. ■

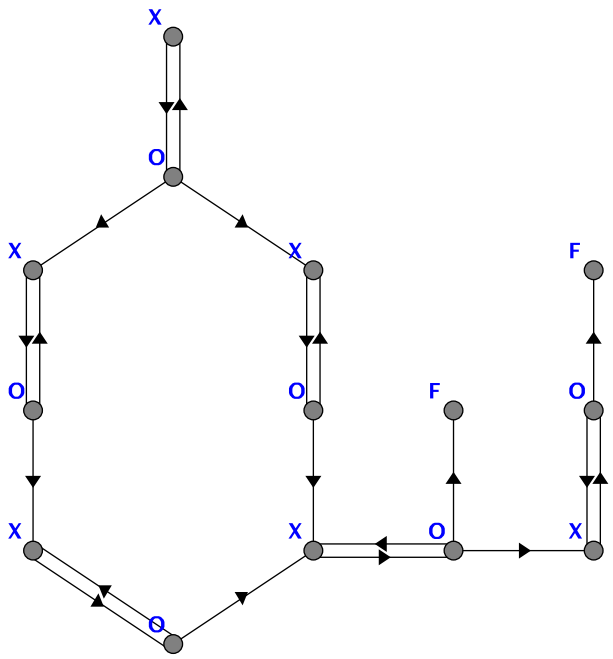


Figure 5.3: FOX result that illustrates Lemma 5.16

Lemma 5.19. [82] Let G be a unicyclic graph with cycle C that can be cracked at node u. Suppose there exists a FOX run on G such that the neighbours y_1 and y_2 of u on C lie in a common S_w for some `F` labelled node w.

Then there exists a subgraph $S_{w'}$ that contains only exactly one of the nodes y_1 and y_2 if and only if C can be cracked at more than one node. □

Proof. If there exists a FOX run on G such that the neighbours y_1 and y_2 of u on C lie in a common S_w for some `F` labelled node w, then with respect to Lemma 4.14 this run can be altered such that $w = y_2$. Note that C is necessarily uncut.

Since y_2 has no outgoing neighbours we see that there exists a subgraph $S_{w'}$ that contains only exactly one of the nodes y_1 and y_2 if and only if there is a second `F` labelled node w' in $\tilde{D}$ that lies on a directed path from y_1. Clearly, w' cannot lie on C. With respect to Lemma 4.11 it is clear that a path from y_1 to w' must exit C at an `O` labelled node v since in $\tilde{D}$ the outgoing neighbours of `X` labelled nodes on C all lie on C themselves.

Let T be the tree that is attached to v and contains w'. Assume that in G node v is adjacent to the node z of T. Employ Lemma 4.14 and alter the FOX run such that $\tilde{D}$ remains unchanged for the nodes of $G - T$ but z becomes `F` labelled. Clearly, this run can then be rearranged such that it first operates on the subtree that cracks C at u and after the cracking continues on the subgraph T for as long as possible. But then z cannot become `F` labelled but instead cracks C at v. ■

Remark 5.20. Given a unicyclic graph G with cycle C that may be cracked at a node u, consider a neighbour y of u on C. If y lies in the subgraph S_w of D for some `F` labelled node w, then u does not belong to S_w. Now apply a FOX run $\mathcal{A}$ to G to get the final digraph $\tilde{D}^{(\mathcal{A})}$ and let $\tilde{K}^{(\mathcal{A})}$ be the subgraph of $\tilde{D}^{(\mathcal{A})}$ induced by the nodes of the component K of $G-u$ that contains y. From the proof of Theorem 5.8 it follows that it is possible to construct a restriction $\mathcal{B}$ of run $\mathcal{A}$ to K such that the result on K is the same, i.e. the digraphs $\tilde{K}^{(\mathcal{A})}$ and $\tilde{K}^{(\mathcal{B})} = \tilde{D}^{(\mathcal{B})}$ are isomorphic. Therefore, for the determination of common subgraphs S_w for the neighbours of u on C as treated in Lemma 5.18 and Lemma 5.19 we may restrict ourselves to K only. □

The final case we need to settle is an uncut cracked cycle C whose size is not a multiple of four. Because of Lemma 5.13 we may restrict ourselves to the case $|C| \equiv 2 \mod 4$.

Let us refine the ideas presented in the proof of Theorem 5.8 and assume that C can be cracked at the node u. Let v be the `X` partner of u.

Then a kernel basis of G can be obtained by suitable linear combinations of the vectors of a given basis B' of $\iota(\ker(G-u))$, enforcing the summation rule at node u. We need to characterise under which restrictions on G this strategy admits the construction of a simply structured kernel basis B. In the following, we tacitly require that the basis B' has been obtained by means of Construction 4.16.

Since C is uncut the neighbours y_1 and y_2 of u on C in G are $\overline{\mathtt{0}}$ labelled. We will distinguish three types of kernel vectors of $G-u$. Depending on whether none, exactly one or both of y_1 and y_2 are assigned nonzero weights by a given vector $x \in \iota(\ker(G-u))$, we say that x is type τ_0, τ_1 or τ_2, respectively.

Next, we make several simple observations:

Observation 5.21. [82] Because of $|C| \equiv 2 \mod 4$ a type τ_2 vector always assigns the same values to y_1 and y_2. In the following, we may therefore assume that B' consists of vectors that assign only 0 or 1 to y_1, y_2. □

Observation 5.22. [82] Consider the component H of $G-u$ that contains v. There exists a kernel vector that does not vanish on v since the restriction of the original FOX run on G to H yields the same node labels, except that v becomes `F` labelled. Therefore, the given basis of $\iota(\ker(G-u))$ always contains a type τ_0 vector. □

Observation 5.23. [82] Since y_1 is $\overline{\mathtt{0}}$ labelled we see that B' always contains at least one type τ_1 or τ_2 vector. □

Observation 5.24. [82] The basis B' contains a type τ_1 vector if and only if for the given FOX run there exists a subgraph S_w that only contains exactly one of the nodes y_1, y_2. Likewise, it contains a type τ_2 vector if and only if y_1 and y_2 lie in a common S_w. □

In view of the previous observation and Lemma 5.18 we see that for a unicyclic graph whose cycle can be cracked at more than one node we may assume a basis B' that does not contain a type τ_2 vector. On the other hand, it may be assumed that a given basis of $\iota(\ker(G-u))$ only contains at most one type τ_2 vector since the difference of two type τ_2 vectors is of type τ_0.

Consequently, the most simple linear combinations of vectors from B' that lead to vectors which have only entries from $\{0, 1, -1\}$ and satisfy the summation rule at node u are of the types $\tau_0 - \tau_0$, $\tau_1 - \tau_0$, $\tau_2 - \tau_0 - \tau_0$, and $\tau_2 - \tau_1 - \tau_0$ (multiple vectors of the same type in an expression are assumed to be distinct). With respect to the previous observations and assumptions it is clear that the construction of a simply structured basis of $\ker G$ is possible if and only if it may succeed using only such linear combinations.

Keeping all this in mind we may reason as follows:

Suppose that B' does not contain a type τ_2 vector. Choose a fixed type τ_0 vector and subtract it from the remaining vectors of B'. Then the difference vectors form a simply structured basis of $\ker G$.

If B' contains a type τ_2 vector, then such a construction can succeed if and only if either B' contains a second type τ_0 vector or a type τ_1 vector.

So, in order to solve the final open case we just need to put together the pieces collected so far:

Theorem 5.25. [82] Let G be a unicyclic graph with uncut cracked cycle C. Let $|C| \equiv 2 \mod 4$. Then G has a simply structured kernel basis if and only if one of the following conditions is satisfied:

1. C can be cracked at more than one node.
2. There exist at least two nodes that may crack C at the same node.

□

Proof. G has a simply structured kernel basis if and only if the construction from the proof of Theorem 5.8 succeeds. Using the results of a suitable FOX run construct a basis B' of $\iota(\ker(G-u))$ as described before. We may assume that B' contains at most one type τ_2 vector.

Then G has a simply structured kernel basis if and only if either B' can be chosen not to contain a type τ_2 vector or to contain a second type τ_0 vector or a type τ_1 vector.

Firstly, according to Lemma 5.18 the occurrence of a type τ_2 vector can be avoided if and only if C can be cracked at more than one node. Secondly, more than one type τ_0 vector exists if and only if there exists a FOX run such that C gets cracked at u, and in D there exist at least two $\overline{\mathtt{0}}$ labelled neighbours of u outside C. Equivalently there exist a least two nodes that may crack C at the same node. And finally, Lemma 5.19 states that a type τ_2 vector is accompanied by a τ_1 vector exactly when we could have avoided the type τ_2 vector in the first place. ■

5.3 Algorithmic, algebraic and structural characterisations

We may sum up the cases considered in the previous sections as follows:

Theorem 5.26. [82] Let G be a unicyclic graph with cycle C. Then G has a simply structured kernel basis unless $|C| \equiv 2 \mod 4$ and C can be cracked by exactly one node. □

Proof. Collect the results from Theorem 5.5, Theorem 5.7, Corollary 5.11, Corollary 5.13, Theorem 5.15, and Theorem 5.25. ■

The criteria we have presented so far in order to achieve a complete characterisation of all unicyclic graphs with simply structured kernel bases are formulated in terms of a particular behaviour of the FOX algorithm. We now seek to give a purely algebraic version.

Lemma 5.27. [82] Let G be a unicyclic graph with cycle C. Let u be a node of C and v one of its neighbours outside C. Further, let T be the tree that is connected to C by the edge uv. Then v may crack C at u if and only if there exists a vector $x \in \ker T$ that does not vanish on v. □

Proof. Suppose that C gets cracked at u by v. Restricting the FOX run to the nodes of T only, we obtain a valid FOX run on T such that v is type `F`. According to Lemma 4.19 there exists a vector $x \in \ker T$ that does not vanish on v.

Conversely, by Lemma 4.19 and Lemma 4.14 there exists a FOX run $\mathcal{A}$ on T such that v is type `F`. This run $\mathcal{A}$ represents a partial FOX run $\mathcal{B}'$ on G. Choose v as the next `X` labelled node since it only is connected to u and by construction has no outgoing neighbours among the other nodes of T. But this means that C gets cracked at u by v. ■

With the help of Lemma 5.27 we may now rephrase Theorem 5.26 as follows:

Theorem 5.28. [82] Let G be a unicyclic graph with cycle C. Let T_i be the trees emanating from C and assume that each tree T_i is attached to C by an edge v_iu_i such that u_i and v_i are nodes of C and T_i, respectively.

Then G has a simply structured kernel basis unless $|C| \equiv 2 \mod 4$ and there exists exactly one index i such that $\ker T_i$ does not completely vanish on v_i. □

Having found an algorithmic and an algebraic characterisation, let us now derive a purely structural one.

Theorem 5.29. [82] Let G be a unicyclic graph with cycle C. Let T_i be the trees emanating from C and assume that each tree T_i is attached to C by an edge v_iu_i such that u_i and v_i are nodes of C and T_i, respectively.

Then G has a simply structured kernel basis unless $|C| \equiv 2 \mod 4$ and there exists exactly one index i such that v_i is not covered by some maximum matching of T_i. □

Proof. Rewrite Theorem 5.28 using Corollary 4.26 and Lemma 4.19. ■

It is interesting to remark that in [21] the inertia of unicyclic graphs is determined. It depends on the circuit size and as well as certain matching properties, just as it is the case in Theorem 5.29. Of course, the actual conditions on circuit size and matchings are quite dissimilar.

Moreover, independently, [66] has determined kernel bases for all unicyclic graphs. These bases depend on matchings of unicyclic graphs. They are given in a way such that, if possible, simply structured bases are chosen, so the results are basically equivalent to those presented here.

6 Distance powers of paths and circuits

We define the r-th distance power $G^{(r)}$ of a given graph G as the graph with the same vertex set as G and two vertices adjacent if and only if their distance in G is at most r.

Distance powers of circuits belong to the important class of circulant graphs. Such graphs are frequently used for modelling redundancies of communication networks. Since circulant graphs reveal strong symmetries a number of interesting results on them have been found [25]. In the following sections, we investigate the classes of path and circuit distance powers.

6.1 Path distance powers

For non-complete path distance powers, a direct consequence of Theorem 3.3 is that only eigenvalues from $\{-3,-2,-1,0,1\}$ may potentially admit simply structured eigenspace bases ($q=4$). We can reduce this set of eigenvalues even further:

Theorem 6.1. Let $d < n-1$. If $P_n^{(d)}$ affords a $\{0,1,-1\}$ valued eigenvector for eigenvalue λ, then λ belongs to the set $\{-2,-1,0,1\}$. □

Proof. Let $v=(v_i)$ be a $\{0,1,-1\}$ valued eigenvector for eigenvalue λ of $P_n^{(d)}$ for given n,d with $d<n-1$. We may assume that $v_1 \in \{0,1\}$. Let A be the canonical adjacency matrix of $P_n^{(d)}$. Create a matrix $B=(b_{ij})$ from $A-c\mathcal{I}$ by successively subtracting row i from row $i-1$ for $i=2,\ldots,n$ and afterwards discarding the n-th row. Then the entries of B are

$$b_{ij}=\begin{cases} -(c+1) & \text{for } j-i=0,\\ c+1 & \text{for } j-i=1,\\ 1 & \text{for } j-i=d+1,\\ -1 & \text{for } j-i=-d,\\ 0 & \text{else.}\end{cases}$$

Clearly, $\mathrm{Eig}(c;P_n^{(d)})$ is a subspace of $\ker B$.

Claim 1: Let $x=(x_i)\in \mathrm{Eig}(\lambda;P_n^{(d)})$ and $x_1=\ldots=x_k=0$ for $1\le k<n$. Then, $x_{k+1}=0$ or $\lambda\in\{-2,-1,0\}$.

If $\lambda=-1$, then there is nothing to show. So let $c\neq -1$ and consider the k-th row of B. Depending on k, different cases may arise:

$$\begin{array}{rll} 0 &= (-1-c)(x_k-x_{k+1})-x_{k+d+1} & =(c+1)x_{k+1}-x_{k+d+1}\\ \text{or } 0 &= x_{k-d}+(-1-c)(x_k-x_{k+1})-x_{k+d+1} & =(c+1)x_{k+1}-x_{k+d+1}\\ \text{or } 0 &= x_{k-d}+(-1-c)(x_k-x_{k+1}) & =(c+1)x_{k+1}\\ \text{or } 0 &= (-1-c)(x_k-x_{k+1}) & =(c+1)x_{k+1}.\end{array}$$

The claim now follows from the fact that $x_i \in \{0, 1, -1\}$.

Claim 2: Let $x = (x_i) \in \mathrm{Eig}(\lambda; P_n^{(d)})$ and $x_1 = \ldots = x_k = 1$ for $1 \leq k < n$. Then, $x_{k+1} = 1$ or $\lambda \in \{-2, -1, 0, 1\}$.

In analogy to the first claim we derive the following cases and check their solutions for $x_i \in \{0, 1, -1\}$:

$$\begin{array}{lll} & 0 = (-1-c)(x_k - x_{k+1}) - x_{k+d+1} & = (c+1)(x_{k+1} - 1) - x_{k+d+1} \\ \text{or} & 0 = x_{k-d} + (-1-c)(x_k - x_{k+1}) - x_{k+d+1} & = -c + (c+1)x_{k+1} - x_{k+d+1} \\ \text{or} & 0 = x_{k-d} + (-1-c)(x_k - x_{k+1}) & = -c + (c+1)x_{k+1} \\ \text{or} & 0 = (-1-c)(x_k - x_{k+1}) & = (c+1)(x_{k+1} - 1). \end{array}$$

Now let $c \notin \{-2, -1, 0, 1\}$. According to the previous claims, either $v = (0, \ldots, 0)^T$ or $v = (1, \ldots, 1)^T$. Both cases are impossible since v is an eigenvector and $P_n^{(d)}$ is non-complete for $d < n - 1$. ■

For each of the eigenvalues $-2, -1, 0, 1$ it is easy to find examples of path distance powers admitting a simply structured eigenspace basis for the chosen eigenvalue.

Let us concentrate on the kernel of a path distance power. In Theorem 6.6 we will characterise which path distance powers $P_n^{(d)}$ with $\frac{n-1}{2} \leq d \leq n - 1$ admit simply structured kernel bases. The condition $\frac{n-1}{2} \leq d \leq n - 1$ means that $P_n^{(d)}$ has at least one vertex that is adjacent to all other vertices. Let us explore the general eigenspace structure of such path distance powers.

Lemma 6.2. [79] Let $\frac{n}{2} \leq d \leq n - 1$. Then the vectors

$$\{(0 \mid e_1 - e_2 \mid 0)^T, (0 \mid e_1 - e_3 \mid 0)^T, \ldots, (0 \mid e_1 - e_s \mid 0)^T\},$$

constitute a basis of $\mathrm{Eig}(-1; P_n^{(d)})$. □

As a direct consequence of Lemma 6.2, we can distinguish between two types of eigenvectors of $P_n^{(d)}$ for $\frac{n}{2} \leq d \leq n - 1$ and $\lambda \neq -1$:

$$v^T = (x \mid 0 \mid y) \qquad \text{type I}$$

and

$$cv^T = (x \mid j \mid y) \qquad \text{type II}$$

for some $c \in \mathbb{R}$, $c \neq 0$. Formally, this distinction remains valid even for $\frac{n-1}{2} \leq d \leq n - 1$.

Lemma 6.3. [79] The linear system

$$\left[\begin{array}{ccccccccc|c} 0 & -1 & -1 & -1 & -1 & -1 & -1 & \dots & -1 & 0 \\ -1 & -1 & -2 & -2 & -2 & -2 & -2 & \dots & -2 & -1 \\ -1 & -2 & -2 & -3 & -3 & -3 & -3 & \dots & -3 & -2 \\ -1 & -2 & -3 & -3 & -4 & -4 & -4 & \dots & -4 & -3 \\ -1 & -2 & -3 & -4 & -4 & -5 & -5 & \dots & -5 & -4 \\ \vdots & \vdots & \vdots & & & \text{etc.} & & & & \vdots \\ -1 & -2 & -3 & & & & & & & -k \end{array}\right]$$

with coefficient matrix of dimension $(k+1) \times (k+1)$ is solvable for $k \geq 0$ if and only if $k \not\equiv 3 \mod 6$.

In this case, the solution is

$$\begin{array}{ll} (\nu, -1+\nu, \dots, 1, \nu, -1+\nu, -1, -\nu, 1-\nu, 1, \nu)^T, \nu \in \mathbb{R} & \text{if } k \equiv 0 \mod 6, \\ (-1, 0, \dots, 0, 1, 1, 0, -1, -1, 0, 1, 1, 0)^T & \text{if } k \equiv 1 \mod 6, \\ (-2, -1, \dots, 1, 2, 1, -1, -2, -1, 1, 2, 1, -1)^T & \text{if } k \equiv 2 \mod 6, \\ (1, 2, \dots, -2, -1, 1, 2, 1, -1, -2, -1, 1, 2)^T & \text{if } k \equiv 4 \mod 6, \\ (0, 1, \dots, -1, 0, 1, 1, 0, -1, -1, 0, 1, 1)^T & \text{if } k \equiv 5 \mod 6. \end{array}$$

□

In the following, let $\overleftarrow{x}$ denote the vector that has the same entries as vector x, but in reverse order.

Lemma 6.4. [79] Let A be adjacency matrix of $P_{2n}^{(n-1)}$, $n \in \mathbb{N}$. Then

$$j \notin \operatorname{im}(\mathcal{J} - A)$$

if and only if $n \equiv 4 \mod 6$. Further, if $(\mathcal{J} - A)v = j$ holds for $v = (v_i)$, then

$$\sum_{i=1}^{2n} v_i = \begin{cases} -2 & \text{if } n \equiv 5 \mod 6, \\ 0 & \text{if } n \equiv 0 \mod 6, \\ 1 & \text{if } n \equiv 1 \mod 6, \\ 2 & \text{if } n \equiv 2 \mod 6, \\ 4 & \text{if } n \equiv 3 \mod 6. \end{cases}$$

For $n \equiv 1 \mod 6$ we have

$$v = (r \mid -r \mid r \mid \dots \mid -r \mid \tilde{r} \mid -r \mid r \mid -r \mid r)^T \tag{6.1}$$

with $r = (1-\nu, 1, \nu)$ and $\tilde{r} = (1-\nu, \nu)$. If $n \not\equiv 1 \mod 6$, then $v = (\overleftarrow{x} \mid x)$ with $x \in \mathbb{R}^n$ being a solution of the system from Lemma 6.3. □

Theorem 6.5. [79] Let $\frac{n-1}{2} \leq d \leq n-1$.

Then $P_n^{(d)}$ is singular if and only if either

1. $n \equiv 1 \mod 12 \ \wedge \ (d = \frac{n+1}{2} \ \vee \ d = \frac{n-1}{2})$ or
2. $n - d \equiv 2 \mod 6$.

In both cases, $\dim \ker P_n^{(d)} = 1$. □

Proof. *(Sketch)* We know that a kernel basis can only contain vectors of type I or II. The existence of a type II kernel vector $(x \mid j_s \mid y)^T$ is equivalent to solving the equations $(\mathcal{J}_{2t} - A(P_{2t}^{(t-1)}))(x \mid y)^T = j$ and $j^T(x \mid y)^T = -(s-1)$ for a vector $(x \mid y) \in \mathbb{R}^{2t}$. So we are looking for a vector as in Lemma 6.4, but with a prescribed component sum. The only valid cases are $s = 3, t \equiv 5 \mod 6$ and $s = 1, t \equiv 0 \mod 6$. Analogously, the existence of a type I vector $(x \mid 0_s \mid y)$ leads equations $(\mathcal{J}_{2t} - A(P_{2t}^{(t-1)}))(x \mid y)^T = j$ and $j^T(x \mid y)^T = 0$ and, hence, to the condition that $t \equiv 1 \mod 6$. The result now follows by virtue of Lemma 6.3 and the fact that the conditions for the existence of the two vector types are mutually exclusive. ■

Theorem 6.6. Let $\frac{n-1}{2} \leq d \leq n-1$. Then $P_n^{(d)}$ admits a simply structured kernel basis if and only if either

1. $n \equiv 1 \mod 12 \ \wedge \ d = \frac{n-1}{2}$ or
2. $n - d \equiv 2 \mod 6$.

□

Proof. Consider the conditions for singularity given in Theorem 6.5. The condition $n \equiv 1 \mod 12 \ \wedge \ d = \frac{n+1}{2}$ is equivalent to $s = 3$, $t \equiv 5 \mod 6$. From the proof of Theorem 6.5 and Lemma 6.4 we see that the kernel is spanned by the vector $(x \mid j_s \mid \overleftarrow{x})^T$ where x is given by the case $k \equiv 4 \mod 6$ in Lemma 6.3. So in this case there exists no simply structured basis. The condition $n \equiv 1 \mod 12 \ \wedge \ d = \frac{n-1}{2}$ is equivalent to $s = 1$, $t \equiv 0 \mod 6$. Here we have the case $k \equiv 5 \mod 6$ in Lemma 6.3, yielding a simply structured basis. Finally, the case $n - d \equiv 2 \mod 6$ is equivalent to $t \equiv 1 \mod 6$. We have $k \equiv 0 \mod 6$ in Lemma 6.3 and obtain a suitable basis vector by choosing $\nu = 1$ so that $j^T(x \mid y)^T = 0$. ■

The case $1 \leq d \leq \frac{n}{2}-1$ remains still open. Computer experiments suggest that most of the kernel bases admit simply structured bases in this case. A counterexample for smallest odd n is the basis

$$\{(-1,1,1,-1,-1,0,2,0,-2,0,1,1,-1,-1,1)^T\}$$

for $n = 15$ and $d = 5$. A counterexample for smallest even n is the basis

$$\{(-1,1,0,-1,2,-1,-1,2,-2,0,2,-2,1,1,-2,1,0,-1,1,0)^T,$$
$$(-1,0,1,-1,1,1,-2,1,0,-2,2,0,-1,2,-1,-1,1,-1,0,1)^T\}$$

for $n = 20$ and $d = 3$.

6.2 Circuit distance powers

In this section we analyse distance powers of circuits. For $d \geq \frac{n-1}{2}$ the graph $C_n^{(d)}$ is complete, so it follows that all eigenspaces of $C_n^{(d)}$ admit simply structured eigenspace bases. So let us consider the non-complete case (where $1 \leq d < \frac{n-1}{2}$). For non-complete circuit distance powers it follows directly from Theorem 3.3 is that only eigenvalues from $\{-3,-2,-1,0,1\}$ may potentially admit simply structured eigenspace bases ($q = 4$). We will analyse for which values of n, d a circuit distance power $C_n^{(d)}$ these eigenvalues can occur.

The next two theorems are reformulations of a single theorem in [80] concerned with the nullity of $C_n^{(d)}$.

Theorem 6.7. The eigenvalues of $C_n^{(d)}$ are exactly

$$\lambda_0 = 2d, \quad \lambda_r = \frac{\sin\left((2d+1)\frac{\varphi}{2}\right)}{\sin\frac{\varphi}{2}} - 1 = \frac{\cos((d+1)\varphi) - \cos(d\varphi)}{\cos\varphi - \cos 0} - 1 \tag{6.2}$$

for $\varphi = 2\pi r/n$ and $r = 1, \ldots, n-1$. □

Proof. From Theorem 2.3 it follows directly that the spectrum of a circuit power $C_n^{(d)}$ is

$$\lambda_r + 1 = \sum_{j=-d}^{d} \omega_n^{rj}, \quad r = 0, \ldots, n-1. \tag{6.3}$$

Clearly, $\lambda_0 = 2d$. Consider the case $1 \leq r \leq n-1$. The right hand side of equation (6.3) can be transformed by means of the following well-known trigonometric identity for the functions $D_q(x)$ of the so-called *Dirichlet kernel* [98]:

$$D_q(x) = \sum_{j=-q}^{q} e^{iqx} = \frac{\sin\left((q+\frac{1}{2})x\right)}{\sin\frac{x}{2}}. \tag{6.4}$$

This yields the first part of the claimed identity for λ_r. The second part follows with the help of the cosine subtraction theorem [12]

$$\cos\alpha - \cos\beta = -2\sin\frac{\alpha+\beta}{2}\sin\frac{\alpha-\beta}{2}.$$

■

Corollary 6.8. A non-complete circuit distance power $C_n^{(d)}$ is singular if and only if there exist integers $1 \leq r < n$ and $l \in \mathbb{N}_0$ such that $dr = ln$ or $2(d+1)r = (2l+1)n$. □

Proof. It follows from equation (6.2) that

$$\lambda_r = 0 \iff \frac{\cos((d+1)\varphi) - \cos(d\varphi)}{\varphi} = \frac{\cos\varphi - \cos 0}{\varphi}. \tag{6.5}$$

In effect, we require the slopes of two particular secant lines of the cosine function to be equal. In this case, due to the nature of the cosine curve there are only two possible constellations for which the slopes are the same (cf. Figure 6.1). Either both secant lines must be apart by a non-vanishing multiple of 2π or their endpoints, if projected onto the same period of the cosine curve, must be point symmetrical with respect to $\frac{\pi}{2}$. The first condition means that $d\varphi = 2\pi l$ and the second yields $(d+1)\varphi = \pi + 2\pi l$. The result now follows by observing Theorem 2.3 and the fact that $\varphi = \frac{2\pi r}{n}$. ■

Corollary 6.9. A non-complete circuit distance power $C_n^{(d)}$ has eigenvalue -2 if and only if there exist integers $1 \leq r < n$ and $l \in \mathbb{N}_0$ such that $(d+1)r = ln$ or $2dr = (2l+1)n$. □

Proof. This is analogous to Corollary 6.8, it just turns out that the slopes are required to have opposite and not the same signs. ■

Corollary 6.10. A non-complete circuit distance power $C_n^{(d)}$ has eigenvalue -1 if and only if there exist integers $1 \leq r < n$ and $l \in \mathbb{N}_0$ such that $(2d+1)r = ln$. □

Proof. For $\lambda_r = -1$ equation (6.2) yields $\cos(d\varphi) = \cos((d+1)\varphi)$. This equation is solvable for $\varphi \in (0, 2\pi)$ if and only if there exist integers r, l with $1 \leq r \leq n-1$, $l \in \mathbb{N}_0$ such that $r = ln$ or $(2d+1)r = ln$. The first condition has no valid solution. ■

Note that from Theorem 2.3 we can easily deduce if the multiplicity of an eigenvalue of $C_n^{(d)}$ is even or odd. Simply verify that $\lambda_r = \lambda_{n/2-r}$ and check the value of $\lambda_{n/2}$.

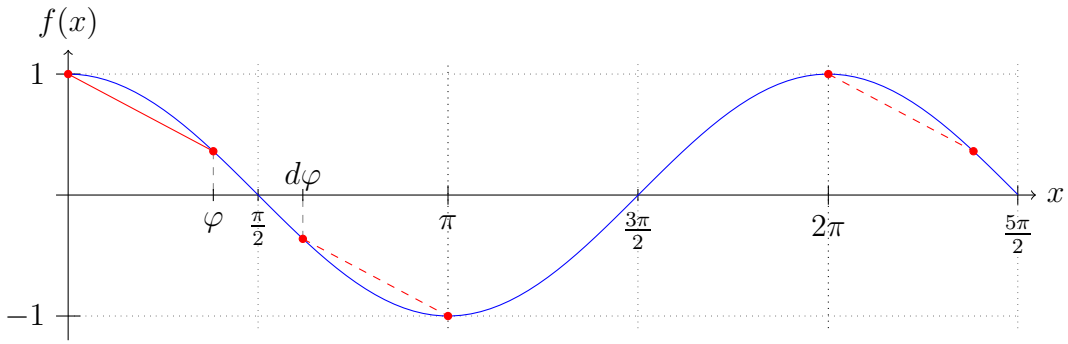


Figure 6.1: Cosine secant lines with same slopes

Observation 6.11. All eigenvalues of $C_n^{(d)}$ have even multiplicity, except $\lambda = -2$ for even n and odd d, $\lambda = 0$ for even n and even d, and $\lambda = 2d$. □

Owing to Theorem 6.7, every valid solution r found in one of the Corollaries 6.8, 6.9, 6.10 contributes to the multiplicity of the respective eigenvalue. So the number of valid solutions r equals the corresponding eigenvalue multiplicity. By revisiting the previous corollaries we can thus precisely determine the actual respective eigenvalue multiplicities.

Let $\mathrm{ord}(p, n)$ denote the order of the prime divisor p with respect to n, i.e.

$$\mathrm{ord}(p, n) = \max\{j \in \mathbb{N}_0 : p^j | n\}.$$

Theorem 6.12. [80] For given $n, d \in \mathbb{N}$ let $g := \gcd(n, d)$ and $h := \gcd(n, d+1)$. If $C_n^{(d)}$ is non-complete, then

$$\dim\ker C_n^{(d)} = \begin{cases} g-1 & \text{if } \mathrm{ord}(2,d+1) \geq \mathrm{ord}(2,n), \\ g+h-1 & \text{if } \mathrm{ord}(2,d+1) < \mathrm{ord}(2,n) \text{ and } 2 \nmid d, \\ g+h-2 & \text{if } \mathrm{ord}(2,d+1) < \mathrm{ord}(2,n) \text{ and } 2 | d. \end{cases}$$

□

Proof. We count the valid solutions in Corollary 6.8.

Claim 1. There exist exactly $g-1$ values of r (with $1 \leq r < n$) such that a solution $l \in \mathbb{N}_0$ exists for $dr = ln$.

Since $\gcd(\frac{n}{g}, \frac{d}{g}) = 1$ we see that the equation $r\frac{d}{g} = l\frac{n}{g}$ has solutions $l \in \mathbb{N}_0$ and $1 \leq r < n$ if and only if $r = j\frac{n}{g}$ with integer j satisfying $1 \leq j \leq g-1$.

CLAIM 2. Let $\operatorname{ord}(2,d+1) < \operatorname{ord}(2,n)$. Then there exist exactly h values of r (with $1 \le r < n$) such that a solution $l' \in \mathbb{N}_0$ exists for $2(d+1)r = (2l'+1)n$.

First note that $\frac{n}{h}$ is even so that $\gcd(\frac{d+1}{h}, \frac{n}{2h}) = 1$. It follows that the equation $r\frac{d+1}{h} = (2l'+1)\frac{n}{2h}$ has solutions $l, r \in \mathbb{N}_0$ exactly for $r = j'\frac{n}{2h}$ and $2l'+1 = j'\frac{d+1}{h}$ with odd integer j' satisfying $1 \le j' < 2h$ (note that $\frac{d+1}{h}$ is odd).

CLAIM 3. Let $\operatorname{ord}(2,d+1) \ge \operatorname{ord}(2,n)$. Then $2(d+1)r = (2l'+1)n$ is not solvable with $1 \le r < n$ and $l' \in \mathbb{N}_0$.

Since $\frac{n}{h}$ is odd it follows that $\gcd(2\frac{d+1}{h}, \frac{n}{h}) = 1$. Consequently, $2r\frac{d+1}{h}$ is even and $(2l'+1)\frac{n}{h}$ is odd so that we cannot solve $2r\frac{d+1}{h} = (2l'+1)\frac{n}{h}$.

CLAIM 4. Let $\operatorname{ord}(2,d+1) < \operatorname{ord}(2,n)$. If d is even, then $r = \frac{n}{2}$ is the only integer $1 \le r < n$ such that simultaneous solutions $l, l' \in \mathbb{N}_0$ can be found for the equations $dr = ln$ and $2(d+1)r = (2l'+1)n$. If d is odd then none such r exists.

From Claims 1 and 2 it follows that necessarily $j_1\frac{n}{g} = (2j_2+1)\frac{n}{2h}$ for integers j_1, j_2 satisfying $1 \le j_1 \le g-1$ and $0 \le j_2 \le h-1$. Equivalently, $j_1 2h = (2j_2+1)g$ must hold. Assume that d is even. Since n is necessarily even it follows that g is even, too. But $\gcd(\frac{g}{2}, h) = 1$ yields that j_1 must be multiple of $\frac{g}{2}$, hence we obtain valid solutions $j_1 = \frac{g}{2}$ and $j_2 = \frac{h-1}{2}$. This yields $r = \frac{n}{2}$. Assume that d is odd. Then g must be odd as well so that $j_1 2h = (2j_2+1)g$ cannot be solved.

In order to finish the proof of the theorem it now suffices to combine the above claims. ■

Theorem 6.13. For given $n, d \in \mathbb{N}$ let $g := \gcd(n,d)$ and $h := \gcd(n,d+1)$. If $C_n^{(d)}$ is non-complete, then

$$\dim \operatorname{Eig}(-2; C_n^{(d)}) = \begin{cases} h-1 & \text{if } \operatorname{ord}(2,d) \ge \operatorname{ord}(2,n), \\ g+h-1 & \text{if } \operatorname{ord}(2,d) < \operatorname{ord}(2,n) \text{ and } 2|d, \\ g+h-2 & \text{if } \operatorname{ord}(2,d) < \operatorname{ord}(2,n) \text{ and } 2\nmid d. \end{cases}$$

□

Proof. In analogy to the proof of Theorem 6.12 we find that $2dr = (2l+1)n$ is solvable for r if and only if $2g|n$, with solutions $\frac{n}{2g} + k\frac{n}{g}$ $(k = 0, \ldots, g-1)$. The equation $(d+1)r = ln$ can be solved for r if and only if $h > 1$, with solutions $k\frac{n}{h}$ $(k = 1, \ldots, h-1)$.

A solution r appears in both sequences at once if and only if there exist integers p, q such that $0 \le p \le g-1$, $1 \le q \le h-1$ and $\frac{n}{2g} + p\frac{n}{g} = q\frac{n}{h}$. The latter condition is equivalent to $(2p+1)h = 2qg$. Since $\gcd(g,h) = 1$ the only possible common solution is $r = \frac{n}{2}$ for $p = \frac{g-1}{2}$, $q = \frac{h}{2}$. This solution exists if and only if h and $g-1$ are odd, which is equivalent to d being odd since n is necessarily even. ■

Theorem 6.14. Let $g = \gcd(2d+1, n)$. If $C_n^{(d)}$ is non-complete, then the multiplicity of -1 as an eigenvalue equals $g - 1$. □

Proof. We count the valid solutions in Corollary 6.10. Let $g = \gcd(2d+1, n)$. Then there exists $q \in \mathbb{N}$ such that $2d+1 = qg$ with $\gcd(q, \frac{n}{g}) = 1$. So $(2d+1)r = ln$ if and only if $rq = l\frac{n}{g}$. Because of $\gcd(q, \frac{n}{g}) = 1$ we get the g solutions $r = \frac{l}{q}\frac{n}{g}$ for $\frac{l}{q} = 1, 2, \ldots, g-1$. ■

Recall that in section 2 we introduced the conjugate transpose F^* of the Fourier matrix. The $(r+1)$-th column of F^* yields a complex eigenvector for eigenvalue λ_r of Theorem 2.3. The same applies to the circuit power eigenvalues given in Theorem 6.7.

Theorem 6.15. If $\lambda \in \{-2, -1, 0\}$ is an eigenvalue of the non-complete graph $C_n^{(d)}$, then the corresponding eigenspace admits a simply structured basis. □

Proof. In the following, let $\mathrm{col}(r)$ denote the $(r+1)$-th column of F^*.

Case $\lambda = 0$: Let $g = \gcd(n, d)$ and $h = \gcd(n, d+1)$. It follows from the proof of Theorem 6.12 that the vectors $u_1, \ldots, u_{g-1}$ with $u_k = \sqrt{n} \cdot \mathrm{col}(kn/g)$ form a basis of a subspace of $\ker C_n^{(d)}$. We will show that the vectors $u'_1, \ldots, u'_{g-1}$ with

$$u'_k = \sum_{m=0}^{n/g-1} e_{k+mg} - e_{g+mg}$$

constitute an alternative (real) basis of this subspace.

Let M be the matrix with columns $u_1, \ldots, u_{g-1}$. Fix some $1 \le \iota \le g-1$ and let M' be the matrix with columns $u_1, \ldots, u_{g-1}, u'_\iota$. Clearly, $\mathrm{rk}\, M' \ge \mathrm{rk}\, M = g-1$. Actually, we have $\mathrm{rk}\, M' = g-1$ since the sum of all row vectors of M' vanishes. To see this, consider the summation of the values in a single column. We have $u_k = (\omega_n^{0kn/g}, \omega_n^{1kn/g}, \omega_n^{2kn/g}, \ldots, \omega_n^{(g-1)kn/g})^T$ so that its component sum is

$$\sum_{m=0}^{g-1} \omega_n^{mkn/g} = \sum_{m=0}^{g-1} \omega_g^{km} \tag{6.6}$$

and therefore a Gaussian period. Because of $1 \le k \le g-1$ we have $g \nmid k$ so that, according to the theory of Gaussian periods [65] [22], the component sum in equation (6.6) vanishes. Moreover, the component sum of u'_ι vanishes, too. As a result, we see that u'_ι is a linear combination of the vectors u_k. Since the vectors $u'_1, \ldots, u'_{g-1}$ are obviously linearly independent, it follows that they are a basis for the space spanned by $u_1, \ldots, u_{g-1}$.

In the case that $\operatorname{ord}(2, d+1) < \operatorname{ord}(2, n)$, equivalently $2h|n$, the vectors $v_1, \ldots, v_h$ with $v_k = \sqrt{n} \cdot \operatorname{col}(kn/h - n/(2h))$ form a basis of another subspace of $\ker C_n^{(d)}$.

A similar argument as for the vectors u'_k shows that the vectors $v'_1, \ldots, v'_h$ with

$$v'_k = \sum_{m=0}^{n/h-1} (-1)^m e_{k+mh}$$

form a basis of the subspace of $\ker C_n^{(d)}$ spanned by $v_1, \ldots, v_h$.

Now consider the cases listed in Theorem 6.12:

- If $\operatorname{ord}(2, d+1) \geq \operatorname{ord}(2, n)$, then $\{u'_1, \ldots, u'_{g-1}\}$ is a basis of $\ker C_n^{(d)}$.
- If $\operatorname{ord}(2, d+1) < \operatorname{ord}(2, n)$ and $2 \nmid d$, then $\{u'_1, \ldots, u'_{g-1}, v'_1, \ldots, v'_h\}$ is a basis of $\ker C_n^{(d)}$.
- If $\operatorname{ord}(2, d+1) < \operatorname{ord}(2, n)$ and $2|d$, then $\{u'_1, \ldots, u'_{g-1}, v'_1, \ldots, v'_h\}$ can be reduced to a basis of $\ker C_n^{(d)}$.

All mentioned bases are simply structured.

Case $\lambda = -2$: Use Theorem 6.13. This case is analogous to case $\lambda = 0$, only with swapped roles of g and h. We have complex subspace basis vectors $u_1, \ldots, u_g$ with $u_k = \sqrt{n} \cdot \operatorname{col}(kn/g - n/(2g))$ and can find real basis vectors $u'_1, \ldots, u'_g$ with

$$u'_k = \sum_{m=0}^{n/g-1} (-1)^m e_{k+mg}$$

for the same subspace. Likewise, we have complex vectors $v_1, \ldots, v_{h-1}$ with $v_k = \sqrt{n} \cdot \operatorname{col}(kn/h)$ and real vectors $v'_1, \ldots, v'_{h-1}$ with

$$v'_k = \sum_{m=0}^{n/h-1} e_{k+mh} - e_{h+mh}.$$

Case $\lambda = -1$: Using Theorem 6.14, we can proceed exactly as in the first part of case $\lambda = 0$, just with $g = \gcd(n, 2d+1)$. We obtain the same complex vectors $u_1, \ldots, u_{g-1}$ and real vectors $u'_1, \ldots, u'_{g-1}$. ■

Example 6.16. According to Theorem 6.15, the vectors

$$\begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 1 \\ 0 \\ -1 \\ \vdots \\ 0 \\ 1 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ -1 \\ 1 \\ 0 \\ -1 \\ 1 \\ 0 \\ -1 \\ \vdots \\ 1 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \\ 1 \\ -1 \\ 0 \\ 1 \\ -1 \\ \vdots \\ 0 \\ 1 \\ -1 \end{pmatrix}$$

constitute a simply structured eigenspace basis of $C_{36}^{(14)}$ for eigenvalue -2. □

There remain two further eigenvalues that may admit simply structured eigenspace bases, namely, -3 and 1. These cases have not been solved yet, but computational experiments suggest the following:

Conjecture 6.17. Let $C_n^{(d)}$ be a non-complete circuit distance power.

1. $C_n^{(d)}$ has eigenvalue -3 if and only if $6|n$ and $d \equiv 4 \mod 6$ for $1 \leq d < \frac{n-1}{2}$. In this case, its multiplicity equals two, with $r \in \{\frac{n}{6}, \frac{5n}{6}\}$ in equation (6.2).
2. $C_n^{(d)}$ has eigenvalue 1 if and only if $6|n$ and $d \equiv 1 \mod 6$ for $1 \leq d < \frac{n-1}{2}$. In this case, its multiplicity equals two, with $r \in \{\frac{n}{6}, \frac{5n}{6}\}$ in equation (6.2).

□

Corollary 6.18. If Conjecture 6.17 holds true, then the eigenvalues -3 and 1 of $C_n^{(d)}$ admit simply structured eigenspace bases. □

Proof. Let $\lambda_r = -3$ or $\lambda_r = 1$ be eigenvalue of $C_n^{(d)}$. It it easily verified that in both cases $\{(1,1,0,-1,-1,0,\ldots)^T, (1,0,-1,-1,0,1,\ldots)^T\}$ is a simply structured basis of the eigenspace for eigenvalue λ_r. ■

As a first step towards proving Conjecture 6.17, it is readily verified with equation (6.2) that $\lambda_r = 1$ for $r = n/6$ or $r = 5n/6$ if and only if $6|n$ and $d \equiv 4 \mod 6$ for $1 \leq d < \frac{n-1}{2}$ (analogously for $\lambda_r = -3$). Moreover, for $\frac{n}{6} < r < \frac{5n}{6}$ we have $\sin\frac{\varphi}{2} > \frac{1}{2}$ so that $|\lambda_r + 1| < 2$. So it remains to show that $\lambda_r \notin \{-3, 1\}$ if only $0 < r < \frac{n}{6}$ (note that r is integer).

7 Product graphs and related classes

In this section we consider product graphs. Such graphs can be created from numerous graph product operations. The names of these operations and the symbols used to denote them vary greatly in common literature. We follow the terminology of [17]. A common feature of all product operations is that the vertex sets of the resulting graphs are Cartesian products of the graphs involved in the respective product operations. Many properties of product graphs can be conveniently expressed in terms of the properties of the original graphs, including eigenvalues and eigenvectors.

7.1 NEPS

Most of the common graph product operations can be classified as NEPS (acronym for "non-complete extended p-sum") operations. Given a set $B \subseteq \{0,1\}^n \backslash \{(0,\ldots,0)\}$ and graphs $G_1,\ldots,G_n$, the NEPS of these graphs with respect to "basis" B is the graph G with vertex set $V(G) = V(G_1) \times \ldots \times V(G_n)$ and edge set $E(G)$ such that $(x_1,\ldots,x_n),(y_1\ldots,y_n) \in V(G)$ are adjacent if and only if there exists an n-tuple $(\beta_1,\ldots,\beta_n) \in B$ such that $x_i = y_i$ whenever $\beta_i = 0$ and $x_i \sim y_i$ in G_i whenever $\beta_i = 1$.

For $n = 2$, commonly used products are

- the direct sum $G_1 + G_2$ with $B = \{(0,1),(1,0)\}$,
- the direct product $G_1 \times G_2$ with $B = \{(1,1)\}$ and
- the strong product $G_1 * G_2$ with $B = \{(0,1),(1,0),(1,1)\}$.

In section 7.2 the far less common left strong product $G_1 \triangleright G_2$ with $B = \{(1,0),(1,1)\}$ will be used.

Since the direct sum commutes it makes sense to extend its definition to arbitrary n. We write rG for the r term sum $G + \ldots + G$. Note that the left strong product, on the other hand, need not commute.

According to [17], the adjacency matrix of a NEPS G of graphs $G_1,\ldots,G_n$ with basis B is

$$A(G) = \sum_{\beta \in B} A(G_1)^{\beta_1} \otimes \ldots \otimes A(G_n)^{\beta_n}.$$

We now cite a well-known result on the eigenvalues of NEPS graphs:

Theorem 7.1. [17] For $i = 1,\ldots,n$, let $\lambda_{i1},\ldots,\lambda_{in_i}$ be the eigenvalues of the graph G_i with n_i vertices with respective linearly independent eigenvectors $x_{i1},\ldots,x_{in_i}$.

Then the eigenvalues of the NEPS G of $G_1, \ldots, G_n$ with basis B are exactly

$$\Lambda_{i_1,\ldots,i_n} = \sum_{\beta \in B} \lambda_{1i_1}^{\beta_1} \cdot \ldots \cdot \lambda_{ni_n}^{\beta_n}$$

with $i_k = 1, \ldots, n_k$ for $k = 1, \ldots, n$.

With each $\Lambda_{i_1,\ldots,i_n}$ associate a vector $x_{i_1,\ldots,i_n} = x_{1i_1} \otimes \ldots \otimes x_{ni_n}$. Then, $x_{i_1,\ldots,i_n}$ is an eigenvector of G for eigenvalue $\Lambda_{i_1,\ldots,i_n}$. Together, these vectors form a complete set of linearly independent eigenvectors of G. □

It follows from Theorem 7.1 that the eigenvalues of the direct sum $G + H$ add up (as already shown in 1947 by Rutherford [77]) whereas for the direct product they are multiplied (hence justifying the names of these operations). The eigenvalues of the left strong product $G_1 \triangleright G_2$ are of the form $\lambda_i(\mu_j + 1)$.

Corollary 7.2. If the eigenvalues of G_1 and G_2 are integer, then also the eigenvalues of every NEPS of G_1 and G_2 are integer. If for every eigenvalue of G_1 and G_2 there exists a simply structured eigenspace basis, then this property also holds for every NEPS of G_1 and G_2. □

Corollary 7.2 generalises a number of previously known results on graph products with integer eigenvalues. The special case that cube-like graphs have only integer eigenvalues dates back to 1975, cf. [61].

More information on NEPS and further generalisations can be found in [17], [16] and [70].

7.2 Cayley graphs and Hamming graphs

Cayley graphs have been an object of study in algebraic graph theory for some time now. They are closely linked to groups, encoding their structures. To be precise, the Cayley graph $\mathrm{Cay}(G, C)$ for a given group G and a subset $C \subseteq G$ has vertex set G and edges exactly between those vertices g, h for which $g^{-1}h \in C$ [10].

We are interested in certain properties of the unitary Cayley graphs $\mathrm{Cay}(\mathbb{Z}_n, U_n)$. Here, U_n denotes the units of $\mathbb{Z}_n$. We can describe the graph $\mathrm{Cay}(\mathbb{Z}_n, U_n)$ as the graph with vertex set $\{1, \ldots, n\}$ and an edge between vertices i and j exactly if $\gcd(i - j, n) = 1$.

Some results on unitary Cayley graphs in particular can be found in [8], [36], [27] and [58]. Note that the unitary Cayley graphs are not to be confused with the coprime graphs of integers, which have a very similar definition (two vertices i, j being adjacent exactly if $\gcd(i, j) = 1$) [42], cf. section 9.

In this section we first expose a link between unitary Cayley graphs and Hamming graphs. The latter are a class of graphs typically related to communication theory. Afterwards we show that all eigenspaces of Hamming graphs and unitary Cayley graphs admit simply structured bases.

Let us first define the class of Hamming graphs. These graphs have evolved over time so that in today's literature the term "Hamming graph" is used for the original class but also for many of the possible generalisations.

The original Hamming graph is defined to model the 1-distance relation in a Hamming scheme. It means that, given an alphabet of m letters, all possible words with r letters from this alphabet are formed. Some definitions even restrict themselves to a binary alphabet. Each of the m^r words is associated with a graph vertex. Two vertices are joined by an edge if their Hamming distance is one, i.e. if their associated words differ in exactly one letter position.

Two immediate generalisations come to mind. Firstly, one can allow each letter position to use its own private alphabet. Secondly, one can model alternative distance relations in the Hamming scheme by joining edges if the Hamming distance is one from a given list K. Our definition of the class of Hamming graphs includes both extensions. We write $\mathrm{Ham}(m_1, \ldots, m_r; K)$ where the respective alphabet sizes are $m_1, \ldots, m_r$ and K a list of positive Hamming distances.

In the 1-distance relation Hamming graph the notions of graph theoretical distance and Hamming distance coincide. Therefore, its r-th distance power is isomorphic to an $1, \ldots, r$-distance relation Hamming graph.

It is not difficult so see that this graph is isomorphic to the direct sum rK_m. Other isomorphies are derived just as easily:

Lemma 7.3. Let $1 \leq d \leq r$. Then,

$$\begin{aligned}
\mathrm{Ham}(m_1, \ldots, m_r; 1) &\simeq K_{m_1} + \ldots + K_{m_r}, \\
\mathrm{Ham}(m_1, \ldots, m_r; 1, \ldots, d) &\simeq (K_{m_1} + \ldots + K_{m_r})^{(d)}, \\
\mathrm{Ham}(m_1, \ldots, m_r; r) &\simeq \overline{\mathrm{Ham}(m_1, \ldots, m_r; 1, \ldots, r-1)}.
\end{aligned}$$

□

Theorem 7.4. Let n be the product of distinct primes $p_1, \ldots, p_r$. Then,

$$\mathrm{Cay}(\mathbb{Z}_n, U_n) \simeq \mathrm{Ham}(p_1, \ldots, p_r; r).$$

□

Proof. Let $n = p_1 \dots p_r$ be a square-free product of primes. We first establish a bijection between the vertex sets of $\mathrm{Cay}(\mathbb{Z}_n, U_n)$ and $\mathrm{Ham}(p_1, \dots, p_r; r)$. Associate with every $x \in \mathbb{Z}_n = \{0, 1, \dots, n-1\}$ the vector $(x_1, \dots, x_r)$ such that each entry x_i is the residue of x modulo p_i. This mapping is injective because otherwise the Chinese Remainder Theorem would be contradicted.

Two vertices $(x_1, \dots, x_r)$ and $(y_1, \dots, y_r)$ in $\mathrm{Ham}(p_1, \dots, p_r; r)$ are adjacent if and only if $x_i - y_i \neq 0$ for all $i = 1, \dots, r$. Viewing the entries as residues we see that this means exactly that $p_i \nmid x_i - y_i$ for all $i = 1, \dots, r$. According to the Chinese Remainder Theorem, this is equivalent to $p_i \nmid x - y$. Consequently, we arrive at $\gcd(x - y, n) = 1$, which is the condition for adjacency in $\mathrm{Cay}(\mathbb{Z}_n, U_n)$. ■

Theorem 7.5. Let $m = p_1^{\alpha_1} \cdot \ldots \cdot p_r^{\alpha_r}$ such that the p_i are the distinct prime divisors of m. Let $n = p_1 \dots p_r$ and $s = \frac{m}{n}$. Then,

$$\mathrm{Cay}(\mathbb{Z}_m, U_m) \simeq \mathrm{Cay}(\mathbb{Z}_n, U_n) \triangleright K_s.$$

□

Proof. Let us revisit the proof of Theorem 7.4. We can choose x e.g. from $n, \dots, 2n-1$ without any impact since the equations are read modulo p_i. Consequently, x and y are adjacent in $\mathrm{Cay}(\mathbb{Z}_m, U_m)$ if and only if for every choice of $j, k \in \{0, \dots, s-1\}$ the vertices $x + jn$ and $y + kn$ are adjacent. So we can extend $\mathrm{Cay}(\mathbb{Z}_n, U_n)$ to $\mathrm{Cay}(\mathbb{Z}_m, U_m)$ by forming the product $\mathrm{Cay}(\mathbb{Z}_n, U_n) \triangleright K_s$ and identifying each vertex (u, v) of the product with the vertex $u + vn$ of $\mathrm{Cay}(\mathbb{Z}_m, U_m)$. Note here that the left strong product with a complete graph K_s in effect means splitting each vertex of the left hand operand graph $s - 1$ times (including neighbourhood). ■

Corollary 7.6. Use the same definitions for m, p_i, s as in Theorem 7.5. Then,

$$\mathrm{Cay}(\mathbb{Z}_m, U_m) \simeq \overline{(K_{p_1} + \ldots + K_{p_r})^{(r-1)}} \triangleright K_s.$$

□

Proof. This follows from Theorem 7.5 together with Theorem 7.4 and Lemma 7.3. ■

A striking property of the adjacency matrix of $\mathrm{Cay}(\mathbb{Z}_n, U_n)$ is that it can be obtained from its first column by repeated downward rotation, so it is a circulant. As noted in section 2, the columns of the conjugate transpose F^* of the $n \times n$ Fourier matrix form a universal set of eigenvectors. However, the entries of these eigenvectors involve

roots of unity and are usually not real. It is not immediately clear how to construct a real basis from these vectors, let alone with further special properties.

We are now concerned with obtaining *real* basis vectors for eigenspaces, in particular with entries from the set $\{0, 1, -1\}$ only. We will show the existence of such bases for the eigenspaces of $\mathrm{Cay}(\mathbb{Z}_n, U_n)$ but beforehand we need some auxiliary results.

Theorem 7.7. Let G be a graph on n vertices and assume that for some positive integer d the distance powers $G^{(1)}, G^{(2)}, \ldots, G^{(d)}$ are simultaneously diagonisable by the eigenvectors $v^{[1]}, \ldots, v^{[n]}$.

Then for every positive integer m and every linearly independent set of m eigenvectors $w^{[j]}$ of K_m the distance powers $H^{(1)}, H^{(2)}, \ldots, H^{(d)}$ with $H = G + K_m$ are simultaneously diagonisable by the eigenvectors $v^{[i]} \otimes w^{[j]}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, s$. □

Proof. According to Theorem 7.1 we only need to assert that every vector $u = v^{[i]} \otimes w^{[j]}$ is also an eigenvector of $H^{(i)}$ for every $1 \leq i \leq d$. In order to test this we can conduct the multiplication of the adjacency matrix of $H^{(i)}$ with a vector u on the graph itself. We assign the k-th component of u as a weight to the k-th vertex of $H^{(i)}$. Then we require that some λ exists such that for every vertex the sum over the weight of its neighbours is equal to λ times its own weight.

Up to isomorphy, the graph H is formed by taking s copies of G and joining all vertices that represent the same vertex in G.

Now fix a pair of vectors $v = v^{[i_0]}$, $w = w^{[j_0]}$. When applying the vector $v \otimes w$ to the vertices of H the vertices of the k-th copy of G are assigned the respective values of v multiplied with the k-th component of w.

Let $1 \leq q \leq d$ and pick a vertex x of $H^{(q)}$. Let y be a neighbour of x and consider a shortest path P from x to y in H. Since the corresponding vertices of the copies of G are mutually connected we may assume that P starts with a number of edges joining the copies of G and then a number of edges within a single copy, totalling length d. This yields two segments P_1 and P_2 of P. Clearly, the length of P_1 must be 0 or 1 because it cannot exceed the diameter of K_m.

So the set of neighbours of x in H consists of the neighbours of x in $G^{(q)}$ (referring to the copy of G that x belongs to) and the neighbours of the vertices corresponding to x in $G^{(q-1)}$ (referring to all the other copies of G in H). By assumption v is an eigenvector for the distance powers $G^{(q)}$ and $G^{(q-1)}$. Let the respective eigenvalues be μ and ν.

Without loss of generality we may make a number of assumptions. The weight of x may be assumed non-zero because otherwise the sum over its neighbour weights vanishes by Theorem 7.1 so that nothing remains to show. Without loss let the

weight of x equal one. We may assume that x lies in the first copy of G in H. Further, let $m \geq 2$ because the case $m = 1$ is trivial. Subsequently, the choice of w can be limited to the vectors $(1, \ldots, 1)$ and $(1, -1, 0, \ldots, 0)$ because these vectors can be extended to a set of linearly independent eigenvectors of K_m such that the added vectors all vanish on x.

Let Σ denote the respective sum over the weights of the neighbours of x. Then we have $\Sigma = \mu + (m-1)\nu$ if $w = (1, \ldots, 1)$ and $\Sigma = \mu - \nu$ if $w = (1, -1, 0, \ldots, 0)$. As we see, the result is independent of the particular choice of x. ■

Observing Corollary 7.2, let us now merge our previous results into our main theorem:

Theorem 7.8. Both Hamming graphs $\mathrm{Ham}(p_1, \ldots, p_r; 1, \ldots, d)$ and the unitary Cayley graphs $\mathrm{Cay}(\mathbb{Z}_n, U_n)$ have only integer eigenvalues. All associated eigenspaces admit simply structured bases. □

Proof. From Corollary 7.6 and Lemma 7.3 it follows that the graphs $\mathrm{Cay}(\mathbb{Z}_n, U_n)$ and $\mathrm{Ham}(p_1, \ldots, p_r; 1, \ldots, d)$ can be constructed from complete graphs by a number of certain graph operations. K_m has a simply structured eigenspace basis for every eigenvalue. Because of Corollary 7.2 the simple basis structure property is preserved for the direct sum and the left strong product. Theorem 2.2 asserts this for the complement of a regular graph. Theorem 7.7 guarantees that taking the distance power of a direct sum of complete graphs also preserves the property. Note that regularity of $(K_{p_1} + \ldots + K_{p_r})^{(r-1)}$ follows from Lemma 7.3 and Theorem 7.4. ■

The fact that unitary Cayley graphs are integral (i.e. that they have only integer eigenvalues) also follows from [58] since they can be expressed as Ramanujan sums (which have only integer values). In [93] a complete characterisation of all integral circulant graphs is achieved. It is interesting to note that circulant graphs can be used to model the behaviour of quantum systems. Such systems exhibit periodicity if any only if the underlying circulant graph is integral [85].

7.3 Sudoku graphs

The recreational game of Sudoku has attained quite some popularity in recent years. A traditional Sudoku puzzle consists of a 3×3 arrangement of square blocks consisting of 3×3 cells each. Each cell may be empty or contain a number ranging from 1 to 9, see Figure 7.1. The aim of the puzzle is to fill the empty cells with numbers from 1 to 9 such that every row, column and block of the puzzle contains all of the numbers $1, \ldots, 9$. A properly set up Sudoku puzzle permits only one unique way of filling the missing numbers. Many different solution techniques exist for Sudoku

puzzles [26]. The game can be generalised to n^4 instead of $3^4 = 81$ cells so that numbers from 1 to n need to be filled in. Let us call these puzzles n-Sudokus.

9	2		1		8	5		7
3		1			6		4	
6	5		4		7		3	2
5	1	6	7		4			8
				6	3		5	1
	9	3	5	8		6	2	
8				1	9	2		
1		7	3					
2	3	9	8	7		4	1	6

Figure 7.1: Example Sudoku puzzle

As a result of Sudoku's general popularity, there has also been an increasing amount of mathematical research on it. In particular, the puzzle exhibits a close connection to graph theory. Given an empty n-Sudoku puzzle, the corresponding Sudoku graph $\mathrm{Sud}(n)$ on n^4 vertices is derived by establishing a one-to-one mapping between the vertices and the cells and adding edges between vertices if and only if the corresponding cells are situated in the same row, column or block. This process is depicted in Figure 7.2.

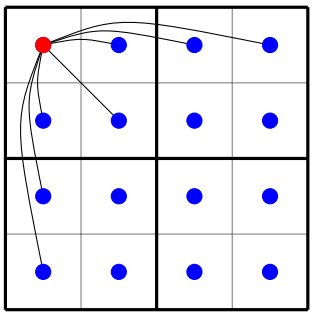


Figure 7.2: Deriving the graph Sud(2) from a 2-Sudoku puzzle

Numbers in the cells of an n-Sudoku puzzle can be interpreted as a vertex colouring of the corresponding Sudoku graph. Hence, the task of solving a Sudoku puzzle is

the mathematical task of extending a partial vertex colouring to a valid n^2-colouring of the entire graph (note that the chromatic number of an n-Sudoku puzzle is n^2 [50]).

Mathematical research on Sudoku has mainly concentrated on aspects of colouring and isomorphism. Typical questions considered are:

- How many completed Sudoku puzzles exist? How many of them are essentially distinct (i.e. how many representative solutions exist from which all other solutions can be constructed by means of permutation or other operations)?
- What is the minimum number of filled cells for which a Sudoku puzzle with a unique solution exists?
- What is the minimum number of filled cells such that for every Sudoku puzzle with that number of filled cells a unique solution exists?
- Given a completed Sudoku puzzle, what is the minimum number of entries to erase such that a non-unique solution is guaranteed to exist?

Further information on the topic can be found in references like [32], [76], [47], [29], and [50].

So far, it appears that no results have been published on the spectral properties of Sudoku graphs. In this section we show that they have only integer eigenvalues and admit simply structured bases for all eigenspaces. The key is the following lemma:

Lemma 7.9. Let $n \in \mathbb{N}$ and $G_1, \ldots, G_4 = K_n$. If G is the NEPS of the G_i for basis $B = \{(0,1,0,1), (1,1,0,0), (0,0,1,1), (1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1)\}$, then $G \simeq \mathrm{Sud}(n)$. □

Proof. We may assume that $V(G_i) = \{1, \ldots, n\}$. Construct a one-to-one mapping between the 4-tuples in $V(G) = \{1, \ldots, n\}^4$ and the cells of the Sudoku grid as follows. For every vertex $v = (a, b, c, d) \in V(G)$, associate with v the cell Γ_v that lies in row number $(a-1)n+b$ and column number $(c-1)n+d$ of the Sudoku grid. Thus, a, c index the vertices and horizontal block number, respectively, whereas b, d index the positions inside the block. So the mapping is clearly one-to-one.

Now fix a vertex $v \in V(G)$ and some $b \in B$ and consider how b selects certain vertices of G as the neighbours of v. We express this in terms of the associated grid cells:

- For $b = (0,1,0,1)$, select all cells in the block of Γ_v that do not lie in the same row or column as Γ_v.

- For $b = (1, 1, 0, 0)$, select all cells in the same column as Γ_v that do not lie in the same block nor at the same relative position inside the block as Γ_v.
- For $b = (1, 0, 0, 0)$, select all cells in the same column as Γ_v that do not lie in the same block but at the same relative position inside the block as Γ_v.
- For $b = (0, 1, 0, 0)$, select all cells in the same column as Γ_v that lie in the same block but not at the same relative position inside the block as Γ_v.

The remaining cases can be resolved in the same manner. Combining the cases, we find that (cf. Figure 7.3)

- subset $S_1 = \{(1, 1, 0, 0), (1, 0, 0, 0), (0, 1, 0, 0)\}$ selects all cells in the same column as Γ_v except Γ_v itself,
- subset $S_2 = \{(0, 0, 1, 1), (0, 0, 1, 0), (0, 0, 0, 1)\}$ selects all cells in the same row as Γ_v except Γ_v itself,
- subset $S_3 = \{(0, 1, 0, 1)\}$ selects all cells of the block of Γ_v not selected by any of the two other subsets, with the exception of Γ_v itself.

But these are exactly the adjacencies of the Sudoku graph. ■

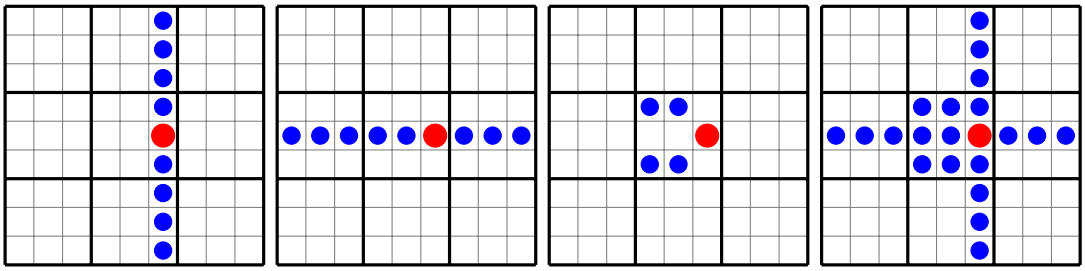


Figure 7.3: Selection of Sudoku cells for basis sets S_1, S_2, S_3, B

Theorem 7.10. All eigenvalues of $\mathrm{Sud}(n)$ are integers. All corresponding eigenspaces admit simply structured bases. □

Proof. Observe that K_n has single eigenvalue $n-1$ with eigenspace basis $\{j_n\}$ and eigenvalue -1 of multiplicity $n-1$ with eigenspace basis $\{e_1-e_2, e_1-e_3, \ldots, e_1-e_n\}$. The result now follows directly from Lemma 7.9 and Corollary 7.2. ■

To complete our analysis of the spectrum of Sudoku graphs, let us determine their exact eigenvalues and eigenvalue multiplicities.

Theorem 7.11. Let $n \geq 2$. Then the spectrum of $\mathrm{Sud}(n)$, in increasing order, is

$$\begin{aligned} -1-n &\ ^{(2n^3-4n^2+2n)},\\ -1 &\ ^{(n^4-2n^3-n^2)},\\ n^2-2n-1 &\ ^{(n^2-2n+1)},\\ n^2-n-1 &\ ^{(2n^2-2n)},\\ 2n^2-2n-1 &\ ^{(2n-2)},\\ 3n^2-2n-1 &\ ^{(1)}. \end{aligned}$$

The graph $\mathrm{Sud}(2)$ has 5 distinct eigenvalues. For $n > 2$, the graph $\mathrm{Sud}(n)$ has 6 distinct eigenvalues. □

Proof. According to Theorem 7.1 and Lemma 7.9, the eigenvalues of $\mathrm{Sud}(n)$ are of the form

$$\Lambda_{\lambda_1,\lambda_2,\lambda_3,\lambda_4} = \lambda_1\lambda_2 + \lambda_2\lambda_4 + \lambda_3\lambda_4 + \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4,$$

where the λ_i are eigenvalues of K_n. Since K_n has only eigenvalues 1 and $k = n-1$ we only need to check 16 cases so that we can conveniently determine the eigenvalues of $\mathrm{Sud}(n)$:

$$\begin{aligned} -2-k &= \Lambda_{k,-1,-1,k} = \Lambda_{-1,k,-1,-1} = \Lambda_{-1,-1,-1,k} = \Lambda_{-1,k,k,-1},\\ -1 &= \Lambda_{k,-1,k,-1} = \Lambda_{-1,-1,k,-1} = \Lambda_{-1,-1,-1,-1} = \Lambda_{k,-1,-1,-1},\\ k^2-2 &= \Lambda_{-1,k,-1,k},\\ k^2+k-1 &= \Lambda_{k,k,-1,-1} = \Lambda_{-1,-1,k,k} = \Lambda_{k,-1,k,k} = \Lambda_{k,k,k,-1},\\ 2k^2+2k-1 &= \Lambda_{-1,k,k,k} = \Lambda_{k,k,-1,k},\\ 3k^2+4k &= \Lambda_{k,k,k,k}. \end{aligned}$$

The respective multiplicities are readily concluded from this analysis and the multiplicities of the eigenvalues of K_n. It is easy to check that for $n \geq 2$ the eigenvalues can be ordered as follows:

$$-1-n < -1 \leq n^2-2n-1 < n^2-n-1 < 2n^2-2n-1 < 3n^2-2n-1.$$

For $n > 2$ this chain of inequalities is strict. ■

8 Cographs

The class of cographs has numerous applications in optimisation, scheduling or even in biology. Consequently, many research results on cographs have been obtained in recent years (see [11] for an overview). The contribution of this chapter is to study the eigenspaces of cographs for eigenvalues 0 and -1, namely, to derive the multiplicities of these eigenvalues and to construct particularly simple eigenspace bases.

The particular eigenvalues 0 and -1 play a role in several areas of algebraic graph theory. For example, in the theory of star partitions the eigenvalues 0 and -1 are special cases (cf. [17], chapter 7). Another interesting result is that singular line graphs of trees can be partitioned into two classes, depending on whether a certain graph has either 0 or -1 as a multiple eigenvalue [88], [44].

For cographs there exist a number of equivalent definitions [11]. For what follows a constructive definition of cographs in terms of split pairs is best suited. Define an operation O_1 on a vertex of a given graph that splits the vertex i.e. it introduces a new vertex with the same neighbourhood. Analogously, an operation O_2 splits a vertex in the same manner, but afterwards adds an edge that connects it with the newly created vertex. The class of cographs is defined as all finite graphs that can be obtained from a single vertex by a series of O_1 and O_2 operations.

In view of the defined operations we call a pair (u, v) of vertices a split pair if their outer neighbourhood is the same, i.e. if $N_G(u)\setminus\{v\} = N_G(v)\setminus\{u\}$. Then a graph is a cograph exactly if it can be reduced to a single vertex by subsequently joining split pairs. This reduction process yields a characteristic series-parallel decomposition tree for every cograph, the so-called cotree.

Every cograph contains at least one split pair, but usually it contains many eligible pairs that can be grouped as follows. Given a graph G with vertex set V, we call a set $M \subseteq V$ a module in G if $N_G(u) \setminus M = N_G(v) \setminus M$ holds for every pair u, v from M. Consequently, a split pair is a module M with $|M| = 2$. A maximum module M with $|M| \geq 2$ such that M is an independent set of vertices in G is called an O_1 cluster. Analogously, if a maximum module M induces a clique in G, then we call M an O_2 cluster. A straightforward argument shows that a vertex cannot be part of more than one cluster.

The number of components of a cograph equals the maximum number of consecutive O_1 operations at the beginning of its construction. The complement $\overline{G}$ of a cograph G is also a cograph, created by the sequence of exactly the opposite O_1 and O_2 operations. Hence, G is connected if and only if $\overline{G}$ is not connected.

We now show that every cograph admits simply structured eigenspace bases for eigenvalues 0 and -1. These bases can be obtained without solving systems of equations. Further, we study the evolution of these eigenspaces when constructing a cograph by repeated vertex split operations.

The next theorem proves the so-called *rank property* of the adjacency matrix of a cograph. This property was first observed by Türker [94] and then proved by Royle [75]:

Theorem 8.1. [75], [94], [13] The rank of a cograph is equal to the number of distinct non-zero rows of its adjacency matrix. □

Remark 8.2. Considering Theorem 8.1, a natural question to ask [75] is whether there exist other common graph classes that have the same rank property of the adjacency matrix as cographs.

Trivially, one can list the path graphs P_{2n} with even number of vertices and the complete graphs K_n.

It is also not difficult to determine exactly which trees have the property that the rank is equal to the number of distinct non-zero rows of the adjacency matrix: The number of distinct non-zero rows of the adjacency matrix of a tree is exactly the number of leaves minus the number of distinct leaf neighbours that aren't leaves themselves. Since the rank of a tree adjacency matrix equals the number of vertices covered by a maximum matching of the tree [19] it follows easily that the rank property holds for exactly those trees for which a matching exists that covers all vertices except the leaves and their non-leaf neighbours.

Moreover, certain generalised line graphs have the rank property, cf. [63]. □

We make use of Theorem 8.1 to state a simply structured basis for the kernel of a cograph:

Theorem 8.3. Let G be a cograph. Then a simply structured basis of $\mathrm{Eig}(0; G)$ can be obtained as follows:

1. For every O_1 cluster M of non-isolated vertices construct $|M| - 1$ vectors by assigning weights
   - 1 to a fixed vertex of M,
   - -1 in turn to exactly one other vertex of M,
   - 0 to all other vertices of G.
2. For every isolated vertex create a unit vector what has weight 1 on the respective isolated vertex.

□

Proof. Since O_1 clusters cannot overlap the constructed vectors are, obviously, linearly independent. Using the summation rule it is readily verified that all the vectors belong to the kernel of G.

Next, observe that the maximal sets of vertices indexed by the redundant rows of the adjacency matrix of G form exactly the O_1 clusters of G and that the all-zero rows correspond to the isolated vertices. Thus, the basis property of the constructed vectors follows directly from Theorem 8.1. ■

Corollary 8.4. Let G be a cograph. Let M_1 be the set of all O_1 cluster vertices of the connected components of G with at least 2 vertices each and let m_1 be the number of such clusters. If by s we denote the number of isolated vertices of G, then

$$\mu(0;G) = |M_1| - m_1 + s.$$

□

The next theorem reveals a fundamental relation between the kernel of a graph and the eigenspace for eigenvalue -1 of its complement.

Theorem 8.5. [81] Let G be a graph with n vertices. Then,

1. $\mathrm{Eig}(0;G) \cap \mathrm{Eig}(-1;\overline{G}) = \{x \in \mathrm{Eig}(0;G) \ : \ {}_j{}^T x = 0\} = \{x \in \mathrm{Eig}(-1;\overline{G}) \ : \ {}_j{}^T x = 0\}$,
2. $|\dim \mathrm{Eig}(0;G) - \dim \mathrm{Eig}(-1;\overline{G})| \leq 1$,
3. $\mathrm{Eig}(0;G) \subset \mathrm{Eig}(-1;\overline{G})$ if $\dim \mathrm{Eig}(0;G) < \dim \mathrm{Eig}(-1;\overline{G})$,
4. $\mathrm{Eig}(0;G) = \mathrm{Eig}(-1;\overline{G})$ if $\dim \mathrm{Eig}(0;G) = \dim \mathrm{Eig}(-1;\overline{G})$,
5. $\mathrm{Eig}(0;G) \supset \mathrm{Eig}(-1;\overline{G})$ if $\dim \mathrm{Eig}(0;G) > \dim \mathrm{Eig}(-1;\overline{G})$.

□

For cographs, the eigenspace inclusion relation described in Theorem 8.5 is not arbitrary:

Theorem 8.6. Let G be a cograph. Then, $\mathrm{Eig}(0;G) \supseteq \mathrm{Eig}(-1;\overline{G})$. □

Proof. We proceed by induction over the number of vertices of G. For $G = K_1$ the result is trivially true since the complement $\overline{G} = K_1$ lacks eigenvalue -1. Let G be a cograph with at least two vertices and assume that the result holds for all cographs with fewer vertices.

If G is disconnected, then by the induction assumption the result holds for each of its components, and hence, by composition, also for G itself.

So let G be connected. We show that for every vector from $\mathrm{Eig}(-1;\overline{G})$ the sum over its components vanishes. Then the result follows from Theorem 8.5.

Assume, to the contrary, that there exists a vector $v \in \mathrm{Eig}(-1;\overline{G})$ with ${}_j{}^T v \neq 0$. It follows by Theorem 8.5 that $\mathrm{Eig}(0;G) \subseteq \mathrm{Eig}(-1;\overline{G})$ so that every vector from the kernel of G has vanishing component sum, in particular the kernel basis vectors of G listed in Theorem 8.3. Hence, according to Theorem 8.5, together with the vector v they form a basis of $\mathrm{Eig}(-1;\overline{G})$. Consequently, we may assume without loss that v vanishes on every O_2 cluster of $\overline{G}$ (recall that the O_2 clusters of $\overline{G}$ are the same as the O_1 clusters of G), except for at most one vertex per cluster.

If there exists an O_2 cluster in $\overline{G}$, then it contains a split pair with a vertex x on which v vanishes. Let $G' = G \setminus \{x\}$ and let $v' = v|_{G'}$ be the restriction of v to G'. Then, ${}_j{}^T v' \neq 0$ and $v' \in \mathrm{Eig}(-1;\overline{G'})$, contradicting the induction assumption.

If there exists no O_2 cluster in $\overline{G}$, then G contains no O_1 cluster but necessarily at least one O_2 split pair. Let G' be the induced subgraph of G obtained by successively joining O_2 split pairs for as long as possible. If $G' = K_1$, then G is a complete graph, for which the result of the theorem is trivially fulfilled. Otherwise, G' contains an O_1 split pair so that $\overline{G'}$ contains an O_2 split pair. Create a vector v' that differs from the null vector only on these O_2 split pair vertices, where it takes values 1 and -1. Clearly, w' is a valid eigenvector of $\overline{G'}$ for eigenvalue -1. Next observe that, by construction, every vertex of $\overline{G}$ that is not a vertex of $\overline{G'}$ is adjacent to either both or none of the mentioned O_2 split pair vertices. So we can trivially extend w' with zeroes to obtain an eigenvector w of $\overline{G}$ for the same eigenvalue since the summation rule still holds. G has no O_1 cluster and, being connected, no isolated vertices. It follows from Theorem 8.1 that the adjacency matrix of G has full rank and further from Theorem 8.5 that $\dim \mathrm{Eig}(0;G) = 0$ and $\dim \mathrm{Eig}(-1;\overline{G}) = 1$. We deduce that w must be a multiple of v. But ${}_j{}^T w = 0$ and ${}_j{}^T \neq 0$, so we arrive at another contradiction. ■

Corollary 8.7. Let G be a cograph. For every O_2 cluster M construct $|M| - 1$ vectors by assigning weights

- 1 to a fixed vertex of M,
- -1 in turn to exactly one other vertex of M,
- 0 to all other vertices of G.

Then the constructed vectors constitute a simply structured basis of $\mathrm{Eig}(-1; G)$. □

Proof. According to Theorem 8.5 and Theorem 8.6 a basis of $\mathrm{Eig}(-1; G)$ is given by taking all vectors of a basis for $\mathrm{Eig}(0; \overline{G})$ that have vanishing component sum. Therefore, the result follows from Theorem 8.3. ■

Corollary 8.8. Let G be a cograph. Let M_2 be the set of all O_2 cluster vertices of G and let m_2 be the number of such clusters. Then,

$$\mu(-1; G) = |M_2| - m_2.$$

□

The inclusion relation given in Theorem 8.6 is very remarkable. Exactly the same property is also exhibited by forests [81]. It would be interesting to find other common graph classes with this property.

As a consequence of Theorem 8.3 and Corollary 8.7, one can easily construct simply structured eigenspace bases for eigenvalues 0 and -1 for *all* graphs created during the successive construction of a cograph using operations O_1 and O_2. In a sense, the eigenspaces bases *evolve* during the construction process, by means of embedding and slight modification:

Corollary 8.9. Let G be a cograph and let B be an eigenspace basis for eigenvalue 0 according to Theorem 8.3. Assume that G' is obtained from G by O_i splitting a vertex v of G. Let v' be the newly created vertex in G'. Let B' initially consist of all vectors of B embedded into G' by setting the weight of v' to zero. Transform the set B' according to the following rules:

- Case $i = 1$: Choose a vector from B' that does not completely vanish on the vertices of the O_1 cluster that v' belongs to. Add a new vector to B' that resembles the chosen vector, except that the weights of the -1 vertex and of v' have been swapped. If no such vector exists, then add to B' a vector that is 1 on v, -1 on v' and zero on all other vertices.
- Case $i = 2$ and v is not an O_1 cluster vertex in G: Leave B' as it is.
- Case $i = 2$ and v belongs to an O_1 cluster in G: If there exists a vector in B' that is 1 on v, then choose a fixed vertex w different from v but in the same O_1 cluster of G and swap the weights of v and w for all vectors in B'. In any case B' now contains exactly one vector that does not vanish on v. Remove it from B'.

Then B' is an eigenspace basis G' for eigenvalue 0 and coincides with a basis obtainable by application of Theorem 8.3 to G'.

A similar eigenspace evolution can be outlined for eigenvalue -1. □

From Corollary 8.9 we can directly derive how the multiplicities of the eigenvalues 0 and -1 evolve under successive O_1 and O_2 operations. Clearly, every O_1 splitting of a vertex increases the dimension of $\mathrm{Eig}(0; G)$. The dimension of $\mathrm{Eig}(-1; G)$ may remain unchanged or drop by one, depending on whether an O_2 cluster has been "hit":

Corollary 8.10. Let G be a connected cograph with at least two vertices and let G' be obtained by a splitting operation on a vertex of G. Then the dimensions of the eigenspaces change according to Table 8.1. □

oper.	dest. vertex type	$\mu(0; G') - \mu(0; G)$	$\mu(-1; G') - \mu(-1; G)$
O_1	non O_2 cluster	1	0
O_1	O_2 cluster	1	-1
O_2	non O_1 cluster	0	1
O_2	O_1 cluster	-1	1

Table 8.1: Cograph eigenspace dimension changes under splitting operation

As remarked earlier, the construction process of a cograph can be described by a series parallel decomposition tree, namely, its cotree. The vertices of the cograph form the leaves of the (rooted and directed) cotree. Every non-leaf vertex of the cotree is labelled type 1 or 2, depending on the operation used to obtain its children.

The cotree T of a cograph $G = (V, E)$ can be obtained in $O(|V| + |E|)$ time, cf. [20], [45]. Using the cotree T, it is straightforward to compute the multiplicities of eigenvalues 0 and -1 of G. Observe that two vertices of G belong to the same cluster (or to the set of isolated vertices) exactly if the corresponding leaves of T have a common parent. Therefore, the clusters and isolated vertices can be determined by a depth first search starting from the root of T. The respective contributions of the vertices to $\mu(0; G)$ and $\mu(-1; G)$ can be calculated using Corollaries 8.4 and 8.8. As a result, we obtain the following algorithm:

Algorithm 8.11. Let T be the cotree of a connected cograph G. If r is the root of T, then $(\mu(0; G), \mu(-1; G)) = \texttt{getMult}(r)$, using the procedure given below.

```
Pair getMult (Node v)
  multZero := 0,  multMinusOne := 0,  leafCount := 0
  for each child w of v
```

```
  if (w is a leaf)
    leafCount := leafCount+1
  else
    (multZero, multMinusOne) += getMult(w)
if (leafCount > 0)
  if (v is part of an O1 cluster)
    if (v has a child that is isolated in G)
      multZero += leafCount
    else
      multZero += leafCount - 1
  else
    multMinusOne += leafCount - 1
return (multZero, multMinusOne)
```

□

To conclude this section, let us show that it is not possible to restrict the set of eigenvalues that admit simply structured eigenspace bases for cographs:

Theorem 8.12. For every integer k there exists a non-regular cograph with eigenvalue k and a corresponding simply structured eigenspace basis. □

Proof. The cases $k = 0$ and $k = -1$ have already been proved.

For $k \geq 1$ consider the graph $G = (K_{k+1} \cup K_{k+1}) \nabla K_1$ with $2k+3$ vertices. Let v be a k-eigenvector, i.e. belong to the null space of the matrix

$$A(G) - k\mathcal{I} = \begin{bmatrix} \mathcal{J}_{k+1} - (k+1)\mathcal{I}_{k+1} & 0 & j_{k+1} \\ 0 & \mathcal{J}_{k+1} - (k+1)\mathcal{I}_{k+1} & j_{k+1} \\ j_{k+1}^T & j_{k+1}^T & -k \end{bmatrix}.$$

Considering differences between any two of the first $k+1$ rows we conclude that the values of v must be the same for all vertices of the first K_{k+1}. By symmetry, the same holds for the second K_{k+1}. It follows that the value of v on the K_1 must be zero. So v is a multiple of a $\{0, 1, -1\}$-vector, cf. Figure 8.1.

For $k \leq -2$ consider the graph $G = K_{-k} \nabla K_{-2k-1}$ with $-3k - 1$ vertices. Consider a vector v from the null space of

$$A(G) - k\mathcal{I} = \begin{bmatrix} \mathcal{J}_{-k} - (k+1)\mathcal{I}_{-k} & \mathcal{J}_{-k,-2k-1} \\ \mathcal{J}_{-2k-1,-k} & -k\mathcal{I}_{-2k-1} \end{bmatrix}.$$

By forming row differences among the first $-k$ rows or among the last $-2k-1$ rows we deduce that v must take the same value on all vertices of the K_{-k} and the same value on all vertices of the K_{-2k-1}. So v is a multiple of a $\{0, 1, -1\}$-vector. An example is shown in Figure 8.2. ■

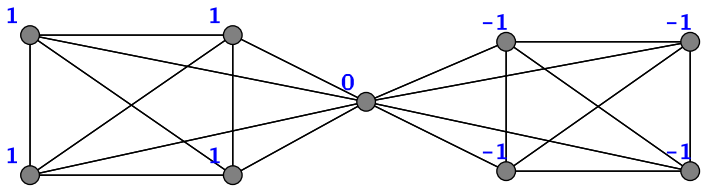


Figure 8.1: Construction for positive eigenvalues ($k = 3$)

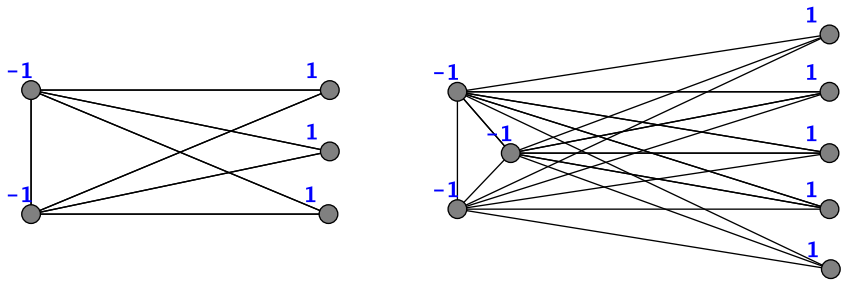


Figure 8.2: Construction for negative eigenvalues ($k = -2$ and $k = -3$)

9 Coprime graphs

In the year 1962 Erdős posed the following question [30]: Given some $k \in \mathbb{N}$, what is the largest integer n such that the set $\{1, \ldots, n\}$ does not contain a subset of size k with mutually coprime elements? Interpreting the set $\{1, \ldots, n\}$ as the vertices of a graph whose edges are given by the relation of pairwise coprimality of these numbers, the problem can be rephrased as a clique problem on a graph.

Formally, for every integer $n > 1$, define the (traditional) coprime graph $\mathrm{TCG}(n)$ as follows. Let $\{1, 2, \ldots, n\}$ be its vertex set and let vertices i, j be adjacent if and only if $\gcd(i, j) = 1$ (i.e. if they are coprime). Obviously, $\mathrm{TCG}(n)$ has a loop at vertex 1. Removing the loop, we obtain the loopless coprime graph $\mathrm{LCG}(n)$.

Several aspects of coprime graphs have been researched, for example finding extremal cliques [30], extremal anti-cliques [2], [3], [4], [1], matchings [72], extremal cycles [31], and complete tripartite subgraphs [84]. There also exist several other graphs on the integers, relating number theory and graph theory [42].

In the following sections, we will determine the nullity of coprime graphs and identify singular coprime graphs admitting simply structured kernel bases.

With regard to what we intend to prove $\mathrm{LCG}(n)$ requires more involved techniques than $\mathrm{TCG}(n)$. For that reason we shall mainly deal with $\mathrm{LCG}(n)$ and comment only in the final section on the corresponding results for $\mathrm{TCG}(n)$, which can be obtained by the same method with less effort.

9.1 Loopless coprime graph

This section deals with the kernel of the loopless coprime graph. Throughout, let $A_n = (a_{ij}) = A(\mathrm{TCG}(n))$ and let a_i denote the i-th row vector of A_n.

9.1.1 Lower bound on the nullity

We denote by

$$\kappa(m) = \prod_{p \in \mathbb{P},\, p \mid m} p$$

the square-free kernel of a positive integer m.

For each square-free integer $k > 1$, the vector $b = (b_1, b_2, \ldots, b_n) \in \mathbb{R}^n$ shall be called k-basic if

$$b_j = \begin{cases} 1 & \text{for } j = k, \\ -1 & \text{for } j = m, \\ 0 & \text{else} \end{cases}$$

for some $m > k$ satisfying $\kappa(m) = k$.

If $b \in \mathbb{R}^n$ is k-basic for some square-free k, we call it a basic vector. The set of all basic vectors $b \in \mathbb{R}^n$ will be denoted by $\mathcal{B}_n$. Let $\nu(n) := |\mathcal{B}_n|$ be the number of basic vectors.

For the next lemma recall that the Moebius function $\mu : \mathbb{N} \to \{-1, 0, 1\}$ is defined by

$$\mu(n) = \begin{cases} 1 & \text{if } n \text{ is square-free with an even number of distinct prime factors,} \\ -1 & \text{if } n \text{ is square-free with an odd number of distinct prime factors,} \\ 0 & \text{if } n \text{ is not square-free.} \end{cases}$$

Lemma 9.1. The number $\nu(n)$ of basic vectors satisfies

$$\nu(n) = n - \sum_{k \le n} |\mu(k)|.$$

□

Proof. We obviously have

$$\begin{aligned} \nu(n) &= |\{b \in \mathbb{R}^n : \ b \text{ is basic}\}| = |\bigcup_{\substack{k=2 \\ \mu(k)\neq 0}}^{n} \{b \in \mathbb{R}^n : \ b \text{ is } k\text{-basic}\}| \\ &= \sum_{\substack{k=2 \\ \mu(k)\neq 0}}^{n} |\{b \in \mathbb{R}^n : \ b \text{ is } k\text{-basic}\}| = \sum_{\substack{k=2 \\ \mu(k)\neq 0}}^{n} |\{k < m \le n : \ \kappa(m) = k\}| \\ &= |\{m \le n : \ \mu(m) = 0\}| = n - \sum_{k \le n} |\mu(k)|. \end{aligned}$$

■

Lemma 9.2. $\mathcal{B}_n$ is linearly independent for every integer $n > 1$. □

Proof. Let $b_1, \ldots, b_{\nu(n)}$ be the elements of $\mathcal{B}_n$. Now let $\lambda_1, \ldots, \lambda_{\nu(n)}$ be real numbers such that

$$\sum_{j=1}^{\nu(n)} \lambda_j b_j = 0. \tag{9.1}$$

For $1 \le j \le \nu(n)$ let $b_j = (b_{j,1}, \ldots, b_{j,n})$. Given some arbitrary but fixed index j_0 with $1 \le j_0 \le \nu(n)$, the vector $b_{j_0} \in \mathcal{B}_n$ is k-basic for some square-free k. It follows

that $b_{j_0,k} = 1$ and $b_{j_0,m} = -1$ for some $k < m \leq n$ with $\kappa(m) = k$. It is obvious that $b_{j,m} = 0$ for all $j \neq j_0$. This implies by equation (9.1) that

$$0 = \sum_{j=1}^{\nu(n)} \lambda_j b_{j,m} = \lambda_{j_0} b_{j_0,m} = -\lambda_{j_0},$$

so that $\lambda_{j_0} = 0$. ■

Lemma 9.3. Let $n > 1$ be an arbitrary integer. Then, $\mathcal{B}_n \subseteq \ker \mathrm{LCG}(n)$. □

Proof. Let $b = (b_1, \ldots, b_n) \in \mathcal{B}_n$. Since b is k-basic for some square-free $k > 1$ it follows that b has entries $b_i = 0$ except $b_k = 1$ and $b_m = -1$ for some m satisfying $k < m \leq n$ and $\kappa(m) = k$. Then we have $a_i b = a_{i,k} - a_{i,m} = 0$ because k and m have the same prime factors and therefore $\gcd(i, k)$ and $\gcd(i, m)$ are both 1 or both greater than 1. This means that b belongs to $\ker \mathrm{LCG}(n)$. ■

We can now derive a lower bound for the nullity of $\mathrm{LCG}(n)$:

Theorem 9.4. For any integer $n > 1$, we have $\dim \ker \mathrm{LCG}(n) \geq \nu(n)$. □

Proof. This follows from Lemma 9.2 and Lemma 9.3. ■

We shall prove in the sequel that in fact $\dim \ker \mathrm{LCG}(n) = \nu(n)$ for most n. This was suggested by numerical calculations. It turns out, however, that there are infinitely many exceptions.

9.1.2 Mertens' function and the kernel

We make use of a truncated version of the Moebius inversion formula, which can be shown in the same fashion as the usual formula (cf. [51], Chapter 6.4, Theorem 4.1):

Lemma 9.5. Let n be a positive integer, and let $f : \{1, 2, \ldots, n\} \to \mathbb{R}$ and $g : \{1, 2, \ldots, n\} \to \mathbb{R}$ be arbitrary functions. Then

$$g(i) = \sum_{d|i} f(d) \quad (1 \leq i \leq n) \quad \Longleftrightarrow \quad f(k) = \sum_{d|k} \mu(d) g\Big(\frac{k}{d}\Big) \quad (1 \leq k \leq n).$$

□

In the sequel, an important role is played by Mertens' well-known function

$$M(n) := \sum_{k=1}^{n} \mu(k).$$

Trivially, $|M(n)| < n$ for all n. The relevance of this function becomes immediately clear from the facts that $M(n) = o(n)$ is equivalent with the prime number theorem and $M(n) = O(n^{\frac{1}{2}+\varepsilon})$ is equivalent with the Riemann Hypothesis. The famous Mertens conjecture from 1897 saying that $|M(n)| < \sqrt{n}$ was disproved by Odlyzko and te Riele [68] in 1985.

It is well known that the summatory function $\varepsilon(n) = \sum_{d|n} \mu(d)$ of the Moebius function satisfies

$$\varepsilon(n) = \begin{cases} 1 & \text{for } n = 1, \\ 0 & \text{for } n > 1. \end{cases}$$

This implies

$$\varepsilon(\gcd(i,j)) = \sum_{\substack{d|i \\ d|j}} \mu(d) = \begin{cases} 1 & \text{for } \gcd(i,j) = 1, \\ 0 & \text{for } \gcd(i,j) > 1. \end{cases} \tag{9.2}$$

Lemma 9.6. Let $n > 1$ be an arbitrary integer. A vector $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ lies in $\ker \mathrm{LCG}(n)$ if and only if

$$(M(n) - 1)\, b_1 = 0 \quad \text{and} \quad \sum_{\substack{j=k \\ j \equiv 0 \bmod k}}^{n} b_j - b_1 = 0$$

for all $2 \leq k \leq n$ with $\mu(k) \neq 0$. □

Proof. From equation (9.2) it follows that

$$a_{i,j} = \varepsilon(\gcd(i,j)) - \gamma_{ij} \tag{9.3}$$

for all $1 \leq i, j \leq n$, where γ_{ij} equals 1 for $i = j = 1$ and 0 otherwise.

For a given vector $b = (b_1, b_2, \ldots, b_n) \in \mathbb{R}^n$ let $f : \{1, 2, \ldots, n\} \to \mathbb{R}$ be defined by

$$f(k) := \mu(k) \sum_{\substack{j=1 \\ j \equiv 0 \bmod k}}^{n} b_j.$$

By equations (9.2) and (9.3) it follows for $1 \le i \le n$ that

$$\begin{aligned} g(i) := \sum_{d|i} f(d) &= \sum_{d|i} \mu(d) \sum_{\substack{j=1 \\ j \equiv 0 \bmod d}}^{n} b_j \\ &= \sum_{j=1}^{n} b_j \sum_{\substack{d|i \\ d|j}} \mu(d) \;=\; \sum_{j=1}^{n} b_j \varepsilon(\gcd(i,j)) \\ &= \sum_{j=1}^{n} a_{i,j} b_j + \gamma_{ij} b_1. \end{aligned} \tag{9.4}$$

A vector $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ lies in $\ker \mathrm{LCG}(n)$ if and only if $\sum_{j=1}^{n} a_{i,j} b_j = 0$ for $1 \le i \le n$. By (9.4) this is equivalent to $g(i) = \gamma_{ij} b_1$ for $1 \le i \le n$. By Lemma 9.5 this means that, for $1 \le k \le n$,

$$f(k) = \sum_{d|k} \mu(d) g\Big(\frac{k}{d}\Big) = \mu(k) g(1) = \mu(k) b_1,$$

so that by the definition of f we have

$$\mu(k) \sum_{\substack{j=1 \\ j \equiv 0 \bmod k}}^{n} b_j = \mu(k) b_1.$$

So far we have shown that $b \in \ker \mathrm{LCG}(n)$ if and only if

$$\sum_{\substack{j=1 \\ j \equiv 0 \bmod k}}^{n} b_j = b_1 \qquad (1 \le k \le n,\ \mu(k) \ne 0). \tag{9.5}$$

We have

$$\begin{aligned} \sum_{\substack{k=2 \\ \mu(k) \ne 0}}^{n} \mu(k) \sum_{\substack{j=1 \\ j \equiv 0 \bmod k}}^{n} b_j &= \sum_{k=2}^{n} \mu(k) \sum_{\substack{j=2 \\ j \equiv 0 \bmod k}}^{n} b_j \\ &= \sum_{j=2}^{n} b_j \sum_{\substack{k=2 \\ k|j}}^{n} \mu(k) = \sum_{j=2}^{n} b_j \sum_{\substack{k=2 \\ k|j}}^{j} \mu(k) \\ &= \sum_{j=2}^{n} b_j (\varepsilon(j) - 1) = -\sum_{j=2}^{n} b_j, \end{aligned} \tag{9.6}$$

and by adding the corresponding equations for $k = 2, \ldots, n$ with $\mu(k) \ne 0$ in (9.5) we obtain

$$-\sum_{j=2}^{n} b_j = \sum_{\substack{k=2 \\ \mu(k) \ne 0}}^{n} \mu(k) \sum_{\substack{j=1 \\ j \equiv 0 \bmod k}}^{n} b_j = \sum_{\substack{k=2 \\ \mu(k) \ne 0}}^{n} \mu(k) b_1 = b_1 \sum_{k=2}^{n} \mu(k) = b_1 (M(n) - 1).$$

The addition of this to the equation for $k = 1$ in (9.5) gives

$$b_1 = \sum_{j=1}^{n} b_j - \sum_{j=2}^{n} b_j = b_1 + b_1(M(n) - 1). \tag{9.7}$$

Replacing the equation for $k = 1$ in (9.5) by equation (9.7) does not change the set of solutions. This completes the proof. ■

9.1.3 Nullity and simply structured kernel bases

Theorem 9.7. For any integer $n > 1$ we have

$$\dim \ker \mathrm{LCG}(n) = \begin{cases} \nu(n) & \text{for } M(n) \neq 1, \\ \nu(n) + 1 & \text{for } M(n) = 1. \end{cases} \tag{9.8}$$

□

Proof. According to Lemma 9.6, a vector $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ lies in $\ker \mathrm{LCG}(n)$ if and only if b satisfies the following homogeneous system of linear equations:

$$\begin{aligned} &\sum_{\substack{j=k \\ j \equiv 0 \bmod k}}^{n} b_j - b_1 = 0 \qquad (2 \leq k \leq n,\ \mu(k) \neq 0), \\ &(M(n) - 1)\, b_1 = 0. \end{aligned} \tag{9.9}$$

Thus we obtain the following detailed system:

$$\left.\begin{array}{ccccccccccccc} b_2 & & +b_4 & & +b_6 & & +b_8 & & +b_{10} & \ldots & -b_1 & = & 0 \\ & b_3 & & & +b_6 & & & +b_9 & & \ldots & -b_1 & = & 0 \\ & & & b_5 & & & & & +b_{10} & \ldots & -b_1 & = & 0 \\ & & & & b_6 & & & & & \ldots & -b_1 & = & 0 \\ & & & & & b_7 & & & & \ldots & -b_1 & = & 0 \\ & & & & & & & & b_{10} & \ldots & -b_1 & = & 0 \\ & & & & & & & & & \ddots & \vdots & = & \vdots \\ & & & & & & & & & & (M(n)-1)b_1 & = & 0 \end{array}\right\} \tag{9.10}$$

Apparently (9.10) is a homogeneous system in row-echelon form with n variables. Hence the rank of the coefficient matrix B_n obviously satisfies

$$\mathrm{rk}\, B_n = \begin{cases} \sum_{k=1}^{n} |\mu(n)| & \text{for } M(n) \neq 1, \\ \sum_{k=1}^{n} |\mu(n)| - 1 & \text{for } M(n) = 1. \end{cases}$$

Consequently $\dim \ker \mathrm{LCG}(n) = n - \mathrm{rk}\, B_n$, so that equation (9.8) follows by Lemma 9.1. ■

Corollary 9.8. $\dim\ker \mathrm{LCG}(n) = (1-\frac{6}{\pi^2})n + O(\sqrt{n})$. □

Proof. It is well known [48] that

$$\sum_{k=1}^{n} |\mu(k)| = \frac{1}{\zeta(2)} n + O(\sqrt{n}) = \frac{6}{\pi^2} n + O(\sqrt{n})$$

so that the result follows by Lemma 9.1. ■

Remark 9.9. Apparently, $\dim\ker \mathrm{LCG}(n)$ depends on the value of $M(n)$, more precisely whether $M(n) = 1$ or not. Results of Pintz and others (cf. [71]) show that $M(n)$ oscillates between $\pm\sqrt{n}$ and, since $|M(n+1) - M(n)| \leq 1$, each value between these bounds is attained infinitely many times. In particular $M(n) = 1$ for infinitely many numbers $n > 1$, the first of these being $n = 94, 97, 98, 99, 100, 146, 147, 148$. □

Theorem 9.10. Let n be an integer satisfying $M(n) \neq 1$. Then:

(i) $\mathcal{B}_n$ is a simply structured basis of $\ker \mathrm{LCG}(n)$.

(ii) For the canonical injection $\iota : \mathbb{R}^n \to \mathbb{R}^{n+1}$ with $\iota(b_1, \ldots, b_n) := (b_1, \ldots, b_n, 0)$ we have $\iota(\ker \mathrm{LCG}(n)) \subseteq \ker \mathrm{LCG}(n+1)$.

□

Proof. The first assertion follows from Theorem 9.7, Lemma 9.1, Lemma 9.2 and Lemma 9.3.

By virtue of the first assertion and by the fact that ι is a homomorphism it suffices to prove the second assertion for each $b \in \mathcal{B}_n$ separately. Each such $b = (b_1, \ldots, b_n)$ satisfies

$$b_j = \begin{cases} 1 & \text{for } \ j = k, \\ -1 & \text{for } \ j = m, \\ 0 & \text{else,} \end{cases}$$

for some square-free $k > 1$ and some $m > k$ with $\kappa(m) = k$.

Let $a'_1, \ldots, a'_{n+1}$ be the row vectors of A_{n+1}. Since A_n is a submatrix of A_{n+1}, we clearly have

$$a'_i \iota(b) = a_i b = 0$$

for $1 \le i \le n$ since $b \in \ker \mathrm{LCG}(n)$. To complete the proof we only have to verify that

$$a'_{n+1}\iota(b) = a_{n+1,k} - a_{n+1,m} = 0.$$

This is in fact true since $\kappa(m) = k$ implies $\gcd(n+1,k) = 1$ if and only if $\gcd(n+1,m) = 1$. ■

Theorems 9.7 and 9.10 imply that in case $M(n) = 1$, apart from the basic vectors in $\mathcal{B}_n$, an additional vector $\tilde{b}_n$ is needed to form a basis of $\ker \mathrm{LCG}(n)$. It seems that $\tilde{b}_n$ cannot be defined in the same easy way as the vectors in $\mathcal{B}_n$.

Let $B' = (b'_{i,j})$ be the $n \times n$ matrix with

$$b'_{i,j} = \begin{cases} 1 & \text{if } \ i|j, \\ 0 & \text{else .} \end{cases}$$

By deleting in B' the rows i with $i = 1$ or $\mu(i) = 0$ as well as the columns j with $j = 1$ or $\mu(j) = 0$ we get an $s_n \times s_n$ matrix $\tilde{B}$ with $s_n := \sum_{k=2}^{n} |\mu(k)|$.

Further, let $\tilde{B}_j$ $(1 \le j \le s_n)$ be the matrix obtained from $\tilde{B}$ by replacing the j-th column by ones.

Theorem 9.11. Let $n > 1$ be an integer satisfying $M(n) = 1$. Then $\mathcal{B}_n \cup \tilde{b}_n$ is a basis of $\ker \mathrm{LCG}(n)$, where $\tilde{b}_n = (\tilde{b}_1, \ldots, \tilde{b}_n)$ is defined by $\tilde{b}_j = \det \tilde{B}_j$. □

Proof. From Lemma 9.1, Lemma 9.2 and Lemma 9.3 we know that $\mathcal{B}_n$ is a set of $\nu(n)$ linearly independent vectors of $\ker \mathrm{LCG}(n)$. By Theorem 9.7, a basis of $\ker \mathrm{LCG}(n)$ requires exactly one more vector in $\ker \mathrm{LCG}(n)$ which has to be linearly independent of $\mathcal{B}_n$. Since $\tilde{b}_1 = 1$, but the first component of all vectors in $\mathcal{B}_n$ is 0, the second condition is obviously satisfied. It remains to show that $\tilde{b}_n \in \ker \mathrm{LCG}(n)$.

Note that the proof of Theorem 9.7 showed that $b \in \ker \mathrm{LCG}(n)$ if and only if $B_n \tilde{b} = 0$, where B_n is the coefficient matrix of (9.10) and $\tilde{b} := (b_2, b_3, \ldots, b_n, b_1)$. Consequently, $\tilde{b}_n$ solves (9.10). By definition, $\tilde{b}_j = 0$ for all j with $\mu(j) = 0$ and $\tilde{b}_1 = 1$. Since we have $M(n) = 1$ by hypothesis, (9.10) is reduced to the system

$$\sum_{\substack{j=k \\ j \equiv 0 \bmod k,\ \mu(j) \neq 0}}^{n} \tilde{b}_j = 1 \quad (2 \le k \le n,\ \mu(k) \neq 0) \tag{9.11}$$

with s_n equations and the same number of variables. Now $\tilde{B}$ is the coefficient matrix of the system (9.11), an upper triangular square matrix with ones on its diagonal. Hence $\det \tilde{B} = 1$, and by Cramer's rule the claimed formula follows. ■

Remark 9.12. Observe that $\tilde{b}_n$ is not a $\{-1, 0, 1\}$-valued vector, so $\mathcal{B}_n \cup \tilde{b}_n$ is not a simply structured basis of $\ker \mathrm{LCG}(n)$. We conjecture that $\ker \mathrm{LCG}(n)$ does not have a simply structured basis at all for $M(n) = 1$. □

9.2 Traditional coprime graph

Let us finally consider the traditional coprime graph $\mathrm{TCG}(n)$. We can derive the following analogues of the results presented in the preceding sections:

Lemma 9.13. Let $n > 1$ be an arbitrary integer. A vector $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ lies in $\ker \mathrm{TCG}(n)$ if and only if

$$b_1 = 0 \quad \text{and} \quad \sum_{\substack{j=k \\ j \equiv 0 \bmod k}}^{n} b_j = 0$$

for all $2 \leq k \leq n$ with $\mu(k) \neq 0$. □

Theorem 9.14. For every integer $n > 1$ we have

$$\dim \ker \mathrm{TCG}(n) = \nu(n) = (1 - \frac{6}{\pi^2})n + O(\sqrt{n}).$$

□

Theorem 9.15. For each positive integer n, we have

(i) $\mathcal{B}_n$ is a simply structured basis of $\ker \mathrm{TCG}(n)$.

(ii) For the canonical injection $\iota : \mathbb{R}^n \to \mathbb{R}^{n+1}$ with $\iota(b_1, \ldots, b_n) := (b_1, \ldots, b_n, 0)$ we have $\iota(\ker \mathrm{TCG}(n)) \subseteq \ker \mathrm{TCG}(n+1)$.

□

The proofs of Lemma 9.13, Theorem 9.14 and Theorem 9.15 are easily obtained by adjusting the proofs of Lemma 9.6, Theorem 9.7 and Theorem 9.10 according to the different value of a_{11} in the adjacency matrix.

10 Outlook

In the preceding chapters we have investigated a number of graph classes with respect to simply structured eigenspace bases. Although substantial results have been obtained, there remain some open questions and many more graph classes to investigate. For example, it would be interesting to characterise all graphs $P_n^{(d)}$ admitting simply structured kernel bases in the case $1 \leq d \leq \frac{n}{2} - 1$. Moreover, computer experiments indicate that the following conjectures are probably true:

1. If n is even and $\dim \operatorname{Eig}(0; G) = 1$, then $P_n^{(d)}$ admits a simply structured kernel basis.
2. Let $\lambda = 1$ be an eigenvalue of $P_n^{(d)}$. Then $d = 1$ and λ is a single eigenvalue that affords a simply structured eigenspace basis.
3. Let $\lambda = 0$ be an eigenvalue of $P_n^{(d)}$ for d even. Then the corresponding eigenspace always affords a simply structured eigenspace basis.
4. Let $\lambda = -2$ be an eigenvalue of $P_n^{(d)}$ for d odd. Then the corresponding eigenspace always affords a simply structured eigenspace basis.
5. $C_n^{(d)}$ has simply structured eigenspace bases for eigenvalues -3 and 1.
6. $\ker \operatorname{LCG}(n)$ does not have a simply structured basis for $M(n) = 1$.
7. Every coprime graph (with or without loop) admits a simply structured eigenspace basis for eigenvalue -1.

The ultimate goal would be to find more general criteria for simply structured bases that eventually merge more and more of the results found for individual graph classes.

A Symbol Index

B List of Figures

C List of Tables

D References

[1] R. Ahlswede and V. Blinovsky. *Maximal sets of numbers not containing $k+1$ pairwise coprimes and having divisors from a specified set of primes.* J. Comb. Theory, Ser. A, Vol. 113, No. 8, pp. 1621–1628, 2006. (cited on page 97)

[2] R. Ahlswede and L.H. Khachatrian. *On extremal sets without coprimes.* Acta Arith., Vol. 66, No. 1, pp. 89–99, 1994. (cited on page 97)

[3] R. Ahlswede and L.H. Khachatrian. *Maximal sets of numbers not containing $k+1$ pairwise coprime integers.* Acta Arith., Vol. 72, No. 1, pp. 77–100, 1995. (cited on page 97)

[4] R. Ahlswede and L.H. Khachatrian. *Sets of integers and quasi-integers with pairwise common divisor.* Acta Arith., Vol. 74, No. 2, pp. 141–153, 1996. (cited on page 97)

[5] S. Akbari, A. Alipour, E. Ghorbani and G. Khosrovshahi. *$\{-1, 0, 1\}$-Basis for the null space of a forest.* Linear Algebra Appl., Vol. 414, pp. 506–511, 2006. (cited on page 4)

[6] S. Akbari, N. Ghareghani, G. B. Khosrovshahi and H. R. Maimani. *The kernels of the incidence matrices of graphs revisited.* Linear Algebra Appl., Vol. 414, pp. 617–625, 2006. (cited on page 4)

[7] S. Akbari and S. J. Kirkland. *On Unimodular Graphs.* International Workshop on Combinatorics II. IPM, Tehran, 2006. (cited on page 51)

[8] P. Berrizbeitia and R. E. Giudici. *On cycles in the sequence of unitary Cayley graphs.* Discrete Math., Vol. 282, pp. 239–243, 2004. (cited on page 80)

[9] J. H. Bevis, G. S. Domke and V. A. Miller. *Ranks of trees and grid graphs.* J. Comb. Math. Comb. Comput., Vol. 18, pp. 109–119, 1995. (cited on pages 3, 24, and 49)

[10] N. Biggs. *Algebraic graph theory.* Cambridge Mathematical Library. Cambridge University Press, Cambridge, second edition, 1993. (cited on pages 4, 7, 9, 54, and 80)

[11] A. Brandstädt, L. Van Bang and J. P. Spinrad. *Graph classes: a survey.* SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999. (cited on page 89)

[12] I. N. Bronstein, K. A. Semendjajew, G. Musiol and H. Mühlig. *Taschenbuch der Mathematik.* Verlag Harri Deutsch, 2001. (cited on page 72)

[13] G. J. Chang, L.-H. Huang and H.-G. Yeh. *On the rank of a cograph.* Linear Algebra Appl., Vol. 429, pp. 601–605, 2008. (cited on page 90)

[14] B. Cheng and B. Lui. *On the nullity of graphs.* Electron. J. Linear Algebra, Vol. 16, pp. 60–67, 2007. (cited on page 3)

[15] L. Collatz and U. Sinogowitz. *Spektren endlicher Grafen.* Abh. Math. Sem. Univ. Hamburg, Vol. 21, pp. 63–77, 1957. (cited on page 1)

[16] D. Cvetković and M. Petrić. *Connectedness of the non-complete extended p-sum of graphs.* Zb. Rad., Prir.-Mat. Fak., Univ. Novom Sadu, Ser. Mat., Vol. 13, pp. 345–352, 1983. (cited on page 80)

[17] D. Cvetković, P. Rowlinson and S. Simić. *Eigenspaces of graphs*, Vol. 66 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, Cambridge, 1997. (cited on pages 7, 79, 80, and 89)

[18] D. M. Cvetković, M. Doob and H. Sachs. *Spectra of graphs.* Johann Ambrosius Barth, Heidelberg, third edition, 1995. Theory and applications. (cited on page 7)

[19] D. M. Cvetković and I. M. Gutman. *The algebraic multiplicity of the number zero in the spectrum of a bipartite graph.* Mat. Vesnik, Vol. 9, No. 24, pp. 141–150, 1972. (cited on pages 3, 23, 24, and 90)

[20] G. Damiand, M. Habib and C. Paul. *A simple paradigm for graph recognition: application to cographs and distance hereditary graphs.* Theoret. Comput. Sci., Vol. 263, pp. 99–111, 2001. (cited on page 94)

[21] S. Daugherty. *The inertia of unicyclic graphs and the implications for closed-shells.* Linear Algebra Appl., Vol. 429, No. 4, pp. 849–858, 2008. (cited on page 65)

[22] H. Davenport. *Multiplicative number theory. 2nd ed. Rev. by Hugh L. Montgomery.* Graduate Texts in Mathematics. Springer, 1980. (cited on page 75)

[23] G. J. Davis, G. S. Domke and C. R. Garner Jr. *4-Circulant Graphs.* Ars Combin., Vol. 65, pp. 97–110, 2000. (cited on page 3)

[24] G. J. Davis, G. S. Domke and C. R. Garner Jr. *Ranks of Line Graphs of Regular Graphs.* JCMCC, Vol. 49, pp. 113–128, 2003. (cited on page 3)

[25] P. J. Davis. *Circulant matrices.* John Wiley & Sons, New York-Chichester-Brisbane, 1979. A Wiley-Interscience Publication, Pure and Applied Mathematics. (cited on pages 7, 10, and 67)

[26] T. Davis. The Mathematics of Sudoku. http://www.geometer.org/mathcircles/sudoku.pdf. (cited on page 85)

[27] I. J. Dejter and R. E. Giudici. *On unitary Cayley graphs.* J. Combin. Math. Combin. Comput., Vol. 18, pp. 121–124, 1995. (cited on page 80)

[28] R. Diestel. *Graph theory.* Graduate Texts in Mathematics, Vol. 173. Springer-Verlag, 1959. (cited on page 7)

[29] C. Elsholtz and A. Mütze. *Sudoku im Mathematikunterricht.* Math. Semesterber., Vol. 39, pp. 69–93, 2007. (cited on page 86)

[30] P. Erdős. *Remarks on number theory. IV: Extremal problems in number theory. I.* Mat. Lapok, Vol. 13, pp. 228–254, 1962. (cited on page 97)

[31] P. Erdős and G. Sárközy. *On cycles in the coprime graph of integers.* Electron. J. Combin., Vol. 4, No. 2, 1997. (cited on page 97)

[32] B. Felgenhauer and F. Jarvis. *Mathematics of Sudoku I.* Math. Spectrum, Vol. 39, pp. 15–22, 2005. (cited on page 86)

[33] M. Fiedler. *Eigenvectors of acyclic matrices.* Czechoslovak Math. J., Vol. 25, No. 4, pp. 607–618, 1975. (cited on pages 33 and 45)

[34] S. Fiorini, I. Gutman and I. Sciriha. *Trees with maximum nullity.* Linear Algebra Appl., Vol. 397, pp. 245–251, 2005. (cited on pages 3 and 24)

[35] G. H. Fricke, S. T. Hedetniemi, D. P. Jacobs and V. Trevisan. *Reducing the adjacency matrix of a tree.* Electron. J. Linear Algebra, Vol. 1, pp. 34–43, 1996. (cited on pages 3, 19, 24, and 49)

[36] E. D. Fuchs. *Longest induced cycles in circulant graphs.* Electron. J. Combin., Vol. 12, No. 1, 2005. (cited on page 80)

[37] P. Funk. *Über die Berechnung der kritischen Drehzahlen bei homogenen und fast homogenen Maschinen.* ZAMM, Vol. 15, pp. 113–120, 1935. (cited on page 1)

[38] F. R. Gantmacher. *The theory of matrices. Vols. 1, 2.* Chelsea Publishing Co., 1959. (cited on page 9)

[39] C. Godsil and G. Royle. *Algebraic graph theory*, Vol. 207 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 2001. (cited on page 7)

[40] C. D. Godsil. *Spectra of trees.* Convexity and graph theory (Jerusalem, 1981), North-Holland, pp. 151–159, 1984. (cited on page 13)

[41] C. D. Godsil. *Inverses of Trees.* Combinatorica, Vol. 5, No. 1, pp. 33–39, 1985. (cited on page 13)

[42] R. L. Graham, M. Grötschel and L. Lovász (eds.). *Handbook of combinatorics. Vol. 1-2.* Elsevier (North-Holland), 1995. (cited on pages 80 and 97)

[43] I. Gutman, S.-L. Lee, Y.-L. Luo and Y.-N. Yeh. *Net Signs of Molecular Graphs: Dependence of Molecular Structure.* Int. J. Quantum Chem., Vol. 49, No. 2, pp. 87–95, 1994. (cited on page 13)

[44] I. Gutman and I. Sciriha. *On the nullity of line graphs of trees.* Discrete Math., Vol. 232, pp. 34–45, 2001. (cited on page 89)

[45] M. Habib and C. Paul. *A new vertex splitting algorithm for cograph recognition.* RR-123800, Rapports internes et prépublications du LaBRI, Univ. Bordeaux, 2000. (cited on page 94)

[46] F. Harary. *Graph theory.* Addison-Wesley Publishing Co., Reading, 1969. (cited on page 7)

[47] F. Harary, W. Slany and O. Verbitsky. *On the Computational Complexity of the Forcing Chromatic Number.* SIAM J. Comput., Vol. 37, No. 1, pp. 1–19, 2007. (cited on page 86)

[48] G.H. Hardy and E.M. Wright. *An introduction to the theory of numbers. 6th ed.* Oxford University Press, Oxford, 2008. (cited on page 103)

[49] F. Hazama. *On the kernels of the incidence matrices of graphs.* Discrete Math., Vol. 254, pp. 165–174, 2002. (cited on page 4)

[50] A. M. Herzberg and M. R. Murty. *Sudoku Squares and Chromatic Polynomials.* Notices AMS, Vol. 6, No. 54, pp. 708–717, 2007. (cited on page 86)

[51] L.K. Hua. *Introduction to number theory.* Springer, 1982. (cited on page 99)

[52] E. Hückel. *Quantentheoretische Beiträge zum Benzolproblem. I. Die Elektronenkonfiguration des Benzols und verwandter Verbindungen.* Z. Phys., Vol. 70, pp. 204–286, 1931. (cited on pages 1 and 2)

[53] W. Imrich and K. S. Zar. *Product Graphs: Structure and Recognition.* John Wiley, 2000. (cited on page 7)

[54] D. P. Jacobs and V. Trevisan. *The Determinant of a Tree's Neighborhood Matrix.* Lin. Algebra Appl., Vol. 256, pp. 235–249, 1997. (cited on page 19)

[55] P. E. John and G. Schild. *Calculating the characteristic polynomial and the eigenvectors of a tree.* Match, No. 34, pp. 217–237, 1996. (cited on pages 3, 19, 23, 24, 30, and 50)

[56] C. R. Johnson and C. M. Saiago. *Estimation of the maximum multiplicity of an eigenvalue in terms of the vertex degrees of the graph of a matrix.* Electron. J. Linear Algebra, Vol. 9, pp. 27–31, 2002. (cited on page 33)

[57] D. Jungnickel. *Graphs, Networks and Algorithms (Algorithms and Computation in Mathematics).* Springer, 2007. (cited on page 7)

[58] W. Klotz and T. Sander. *Some properties of unitary Cayley graphs.* Electron. J. Combin., Vol. 14, No. 1, 2007. (cited on pages 4, 80, and 84)

[59] A. Leal-Duarte and C. R. Johnson. *On the minimum number of distinct eigenvalues for a symmetric matrix whose graph is a given tree.* Math. Inequal. Appl., Vol. 5, No. 2, pp. 175–180, 2002. (cited on page 33)

[60] J. Li. *The determinant of a unicyclic graph's neighborhood matrix.* Linear Algebra Appl., Vol. 394, pp. 201–216, 2005. (cited on page 3)

[61] L. Lovász. *Spectra of graphs with transitive groups.* Periodica Math. Hung., Vol. 6, No. 2, pp. 191–195, 1975. (cited on page 80)

[62] R. B. Mallion. *Some chemical applications of the eigenvalues and eigenvectors of certain finite, planar graphs.* Applications of combinatorics (Buckinghamshire, 1981), Vol. 6 of *Shiva Math. Ser.*, pp. 87–114, 1982. (cited on pages 1 and 2)

[63] M. C. Marino, I. Sciriha, S. K. Simić and D. V. Tošić. *More about singular line graphs of trees.* Publications de l'Institut Mathmatique (Beograd), Vol. 79, No. 93, pp. 70–85, 2006. (cited on pages 3 and 90)

[64] J. J. McDonald. *Partly zero eigenvectors and matrices that satisfy $Au = b$.* Linear and Multilinear Algebra, Vol. 33, pp. 163–170, 1993. (cited on page 33)

[65] T. Nagell. *Introduction to number theory.* Wiley, New York, 1951. (cited on page 75)

[66] M. Nath and B. K. Sarma. *On the null-spaces of acyclic and unicyclic singular graphs.* Linear Algebra Appl., Vol. 427, pp. 42–54, 2007. (cited on pages 3, 4, and 65)

[67] P. Nylen. *Null space structure of tree-patterned matrices.* Linear Algebra Appl., Vol. 279, pp. 153–161, 1998. (cited on pages 32–34 and 45)

[68] A.M. Odlyzko and H.J.J. te Riele. *Disproof of the Mertens conjecture.* J. Reine Angew. Math., Vol. 357, pp. 138–160, 1985. (cited on page 100)

[69] S. Pavlíková and J. Krč-Jediný. *On the inverse and the dual index of a tree.* Linear and Multilinear Algebra, Vol. 28, pp. 93–109, 1990. (cited on page 13)

[70] M. Petrić. *Connectedness of the generalized direct product of digraphs.* Univ. Beograd. Publ. Elektrotehn. Fak. (Ser. Mat.), Vol. 6, pp. 30–38, 1995. (cited on page 80)

[71] J. Pintz. *Oscillatory properties of $M(x) = \sum_{n \leq x} \mu(n)$. III.* Acta Arith., Vol. 43, pp. 105–113, 1984. (cited on page 103)

[72] C. Pomerance and J.L. Selfridge. *Proof of D. J. Newman's coprime mapping conjecture.* Mathematika, Vol. 27, pp. 69–83, 1980. (cited on page 97)

[73] T. Pöschl and L. Collatz. *Über die Berechnung und Darstellung der Eigenfrequenzen homogener Maschinen mit Zusatzdrehmassen.* ZAMM, Vol. 18, pp. 186–194, 1938. (cited on page 1)

[74] D. L. Powers. *Graph partitioning by eigenvectors.* Linear Algebra Appl., Vol. 101, pp. 121–133, 1988. (cited on pages 3 and 33)

[75] G. F. Royle. *The Rank of a Cograph.* Electron. J. Comb., Vol. 10, No. 4, 2003. (cited on page 90)

[76] E. Russell and F. Jarvis. *Mathematics of Sudoku II.* Math. Spectrum, Vol. 39, pp. 54–58, 2005. (cited on page 86)

[77] D. E. Rutherford. *Some continuant determinants arising in physics and chemistry.* Proc. Roy. Soc. Edinburgh. Sect. A., Vol. 62, pp. 229–236, 1947. (cited on pages 1 and 80)

[78] J. W. Sander and T. Sander. *On Simply Structured Bases of Tree Kernels.* AKCE J. Graphs. Combin., Vol. 2, No. 1, pp. 45–56, 2005. (cited on pages 3, 4, 15–24, 27, 29, and 49)

[79] T. Sander. *Eigenspace Structure of Certain Graph Classes.* Ph. D. Thesis, TU Clausthal, 2004. (cited on pages 4, 13, and 68–70)

[80] T. Sander. *Singular distance powers of circuits.* Tokyo J. Math., Vol. 4, No. 1, pp. 491–498, 2007. (cited on pages 3, 71, and 73)

[81] T. Sander. *Inclusion relations of certain graph eigenspaces.* Rocky Mt. J. Math., to appear. (cited on pages 91 and 93)

[82] T. Sander and J. W. Sander. *On Simply Structured Kernel Bases of Unicyclic Graphs.* AKCE J. Graphs. Combin., Vol. 4, No. 1, pp. 61–82, 2007. (cited on pages 4, 18, 20, 25, 26, and 52–65)

[83] T. Sander and J. W. Sander. *Tree decomposition by eigenvectors.* Linear Algebra Appl., 2008. DOI: 10.1016/j.laa.2008.07.015. (cited on pages 34–36, 38–41, and 45–48)

[84] G. Sárközy. *Complete tripartite subgraphs in the coprime graph of integers.* Discrete Math., Vol. 202, No. 1-3, pp. 227–238, 1999. (cited on page 97)

[85] N. Saxena, S. Severini and I.E. Shparlinski. *Parameters of integral circulant graphs and periodic quantum dynamics.* Int. J. Quantum Inf., Vol. 5, No. 3, pp. 417–430, 2007. (cited on page 84)

[86] C.H. Schilling, D. Letscher and B.O. Palsson. *Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspectiv.* J. Theor. Biol., Vol. 203, No. 3, pp. 229–248, 2000. (cited on page 2)

[87] I. Sciriha. *On the construction of graphs of nullity one.* Discrete Math., Vol. 181, pp. 193–211, 1998. (cited on pages 3 and 35)

[88] I. Sciriha. *The two classes of singular line graphs of trees.* 5th Workshop on Combinatorics (Messina, 1999), Rend. Sem. Mat. Messina Ser. II, Vol. 5, No. 21, pp. 167–180, 1999. (cited on page 89)

[89] I. Sciriha. *A characterization of singular graphs.* Electron. J. Linear Algebra, Vol. 16, pp. 451–462, 2007. (cited on pages 3 and 27)

[90] I. Sciriha. *Coalesced and Embedded Nut Graphs in Singular Graphs.* Ars Math. Contemp., Vol. 1, No. 1, pp. 20–31, 2008. (cited on page 2–4)

[91] I. Sciriha and I. Gutman. *Nut Graphs: Maximally Extending Cores.* Util. Math., Vol. 54, pp. 257–272, 1998. (cited on pages 3 and 35)

[92] I. Sciriha and I. Gutman. *Minimal configuration trees.* Linear Multilinear Algebra, Vol. 54, No. 2, pp. 141–145, 2006. (cited on page 3)

[93] W. So. *Integral circulant graphs.* Discrete Math., Vol. 306, No. 1, pp. 153–158, 2006. (cited on page 84)

[94] B. Türker. *Graph Laplacians and Nodal Domains.* Ph. D. Thesis, Universität Wien, Austria, 2003. (cited on page 90)

[95] W.T. Tutte. *The factors of graphs.* Can. J. Math., Vol. 4, pp. 314–328, 1952. (cited on page 1)

[96] R. H. Villarreal. *Rees algebras of edge ideals.* Commun. Algebra, Vol. 39, pp. 3513–3524, 1995. (cited on page 4)

[97] T. Xuezhong and B. Liu. *On the nullity of unicyclic graphs.* Linear Algebra Appl., Vol. 408, pp. 212–220, 2005. (cited on page 3)

[98] A. Zygmund. *Trigonometrical series. 2nd ed.* Dover Publications, New York, 1955. (cited on page 71)