To the University of Wyoming:

The members of the Committee approve the dissertation of Keivan Hassani Monfared presented on July 10, 2014.

Bryan L. Shader, Chairperson

John M. Hitchcock, External Department Member

Farhad Jafari

G. Eric Moorhouse

Jason Williford

APPROVED:

Farhad Jafari, Head, Mathematics Department

Paula M. Lutz, Dean, Arts and Science

Hassani Monfared, Keivan, The Jacobian Method: The Art of Finding More Needles in Nearby Haystacks, Ph.D., Mathematics Department, August, 2014.

Abstract: A method called the Jacobian method is developed, and it is used to solve three fundamental structured inverse eigenvalue problems (SIEP's). The common strategy is to prove the existence of a solution for trees, then to show that these solutions are generic, and then use the Implicit Function Theorem to extend it to connected graphs.

To illustrate this method, it is shown that for any given set of n distinct real numbers Λ and any graph G on n vertices there is a real symmetric matrix A whose graph is G and its spectrum is Λ (Theorem 2.3.4). Then a geometric interpretation of it is provided, in which the notion of a generic solution is explained using transverse intersection of manifolds.

The three fundamental SIEP's are as follows:

- (The λ-μ-SIEP) A result of A.L. Duarte which asserts that for any given tree T on n vertices, a fixed vertex w, a set of n distinct real numbers Λ, and a set of $n-1$ distinct real numbers M, where M strictly interlaces Λ, there is a real symmetric matrix A whose graph is T, its spectrum is Λ, and the spectrum of $A(w)$ is M (Theorem 3.1.1). The Jacobian method is used to show that a similar result holds for connected graphs (Theorem 3.3.1).

- (The λ-τ-SIEP) It is shown that for any given tree G on n vertices and two fixed vertices r and s, a set of n distinct real numbers Λ, and a set of $n-2$ distinct real numbers T, where T and Λ satisfy the strict second order Cauchy interlacing inequalities, there is a real symmetric matrix A whose graph is G, its spectrum is Λ and the spectrum of $A(\{r, s\})$ is T, provided some necessary combinatorial conditions are satisfied (Theorems 4.2.1 and 4.4.2). Then the Jacobian method is used to show that a similar result holds for connected graphs (Theorems 4.3.4 and 4.4.4). Furthermore, the mentioned results are used in order to solve problems related to perturbing one or two diagonal entries of a matrix so that the eigenvalues of the new matrix change as prescribed (Theorems 4.5.5 and 4.5.6).

- (The nowhere-zero eigenbasis SIEP) It is shown that for any given tree T on n vertices and a set of n distinct real numbers Λ, there is a matrix whose graph is T and its spectrum is Λ such that none of the eigenvectors of the matrix have a zero entry (Theorem 5.1.8). Then that result is extended to any connected graph (Theorem 5.1.9). In the proof, both results from the λ-μ-SIEP and the λ-τ-SIEP are used.

Also, a series of problems that could be solved using the Jacobian method are proposed and discussed.

THE JACOBIAN METHOD: THE ART OF FINDING MORE NEEDLES IN NEARBY HAYSTACKS

by

Keivan Hassani Monfared, Master of Science

A dissertation submitted to the
Mathematics Department
and the
University of Wyoming
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
in
MATHEMATICS

Laramie, Wyoming
August 2014

To my parents, Tayebeh and Hassan

Contents

List of Figures

List of Computer Programs

Acknowledgments

First and foremost I offer my sincerest gratitude to my advisor, Professor Bryan L. Shader, who has supported me throughout the past five years with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my Ph.D. degree to his encouragement and effort and without him this dissertation, too, would not have been completed or written. One simply could not wish for a better and friendlier supervisor.

I am also grateful for Professor John M. Hitchcock's support and help. Professor Farhad Jafari has always been a great help and a great friend, who has supported me throughout every single step of my graduate studies at the University of Wyoming. I would also like to thank Professor G. Eric Moorhouse, who helped me to understand many concepts related to my research, and also Professor Jason Williford, who greatly influenced my research by helping me in various ways. I am also thankful for all the things that I have learned from the faculty members of the mathematics department of the University of Wyoming, especially Professor Chanyoung Shader, Professor Chris Hall, Professor Tyrrell McAllister, and Professor Charlie Angevine.

Nonetheless, life would not be as easy as it is without help, support and assistance of Ms. Beth Buskirk, who I always will be grateful for.

Ehssan Khanmohammadi has been a great friend, colleague, and essential critic of my works and thoughts, whom I discussed a lot of details of my research, and learned lots of math. Without him I would have probably been doing something else other than mathematics these days. There would be no way to thank him for all he has done for me.

I have had numerous fruitful discussions with Dr. Sudipta Mallik, and Curtis Nelson which greatly influenced my work, and made it more meaningful.

The Department of Mathematics has provided the support and equipment I have needed to produce and complete my dissertation. Furthermore, the conferences, seminars and colloquia that the department has held or supported my travel to them have been great motivations for my work.

I am also indebted to Professor Parviz Shahriari, who essentially taught me what mathematics really is and why it is important, by writing books and articles that I started reading in middle school.

Finally, I thank my parents for supporting me throughout all of my studies, and all of my life, and also all of my beloved friends who continue to give meaning to my life.

Keivan Hassani Monfared

University of Wyoming
August 2014

Chapter 1

Introduction

> "Suppose that we ski down a mountain trail. As long as we neglect friction and know the undulations in the trail, we can calculate exactly the time it will take to travel from a point on the mountain side to the valley below. This is a direct problem. It was difficult for Galileo's contemporaries. It is, however, now old hat and what is of current deeper interest is the following problem. If we start skiing from different places on the slope and on each occasion we time our arrival at a fixed place on the valley floor, how can we calculate the topographic profile of undulations of the trail? This is the inverse problem. It is certainly a practical problem. It is also challenging and difficult and, indeed, in the general sense, it has no unique solution."

The author of the above, Bolt, uses this illustration to describe what an inverse problem is [1]. Chu and Golub in their 2002 paper [2] 'Structured Inverse Eigenvalue Problems' express that eigenvalues play an enigmatic yet important role in nature. Additionally, they mention: "The process of analysing and deriving the spectral information and, hence, inferring the dynamical behavior of a system from *a priori* known physical parameters such as mass, length, elasticity, inductance, capacitance, and so on is referred to as a *direct* problem. The *inverse* problem then is to validate, determine, or estimate the parameters of the system according to its observed or expected behavior."

Furthermore, inverse eigenvalue problems (IEP's) ask whether or not there is a matrix in a certain family of matrices that has some specific eigenvalues. If such matrix exists, the next natural step is to find one, provide an algorithm to find one, or characterize all the matrices with the desired property. The easiest inverse eigenvalue problem is to find an $n \times n$ matrix whose eigenvalues are given by numbers $\lambda_1, \lambda_2, \ldots, \lambda_n$. An answer is a diagonal matrix whose diagonal entries are $\lambda_1, \lambda_2, \ldots, \lambda_n$. Let A denote the diagonal matrix

$$A = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}.$$

Clearly, for any invertible matrix P, PAP^{-1} is a solution to the mentioned inverse eigenvalue problem.

The question becomes interesting when there are some other restrictions on the solution matrix. In this dissertation we will concentrate on the case that the 'zero-nonzero pattern' of the matrix is specified. Such problems are a portion of the structured inverse eigenvalue problems (SIEP's) [2]. We mainly consider the case where the problem asks about the existence of a symmetric matrix, and the zero-nonzero pattern of the matrix is described by a graph. That is, for a given simple graph G on n vertices $1, 2, \ldots, n$, we consider the symmetric matrices $A = \begin{bmatrix} A_{ij} \end{bmatrix}$ that for all $i \neq j$ we have $A_{ij} \neq 0$ if and only if vertex i is adjacent to vertex j in G. Note that, this does not restrict the diagonal entries of A. In his 1989 paper [3], Duarte discusses why is it necessary to not restrict the diagonal entries. More precisely, one wants to define a function for each entry of the matrix A in terms of λ_i's, so that the functions map each set of λ's to a matrix with those eigenvalues. These functions on the diagonal entries cannot be constantly zero, if one wants to be able to achieve *any* set of eigenvalues. For example, if all the prescribed eigenvalues are positive, that is the solution matrix is a positive-definite (PD) matrix, then by definition for each nonzero vector x of length n, $x^T Ax > 0$. Now let $x = e_i$. Then $e_i^T Ae_i$ is the i-th diagonal entry of A. So, all the diagonal entries should be positive.

1.1 Definitions and Notation

Throughout, $M_n(\mathbb{R})$ denotes the set of all $n \times n$ real matrices. Let $A \in M_n(\mathbb{R})$. Then A^T denotes the $n \times n$ *transpose* of A, that is $(A^T)_{ij} = A_{ji}$ for all $j = 1, 2, \ldots, n$ and all $i = 1, 2, \ldots, n$. We say A is *symmetric* when $A^T = A$, and we denote the set of all real symmetric matrices by $\mathrm{Sym}_n(\mathbb{R})$. The matrix A is said to be *orthogonal* if $AA^T = I$, the identity matrix. The set of all $n \times n$ real orthogonal matrices is denoted by $O_n(\mathbb{R})$.

If $A\boldsymbol{v} = \lambda\boldsymbol{v}$ for some nonzero vector $\boldsymbol{v}$ and some number λ, then λ is called an *eigenvalue* of A and $\boldsymbol{v}$ is called an *eigenvector* corresponding to λ. The pair $(\lambda, \boldsymbol{v})$ is sometimes called an *eigenpair* of A. For a matrix $A \in M_n(\mathbb{R})$, the polynomial $C_A(x) = \det(xI - A)$ is the *characteristic polynomial* of A, and the roots of this polynomial are the eigenvalues of A. If all the eigenvalues of a real symmetric matrix A are positive (non-negative), then A is called a *positive-definite* (respectively, *positive-semidefinite* or *non-negative-definite*) matrix. The vector of size appropriate to the context with a 1 in position i and all other entries 0 is denoted by e_i. Furthermore, E_{ij} is the $m \times n$ matrix with (i, j) entry equal to 1 and all other entries are zero, and m and n appropriately chosen according to the context. We denote the (multi)set of all the eigenvalues of A by $\sigma(A)$, and it is called the *spectrum* of A. If the spectrum of a matrix A is $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$, then we say A *realizes* Λ. The number of independent rows of a matrix A is the row rank of A. If A is an $m \times n$ matrix with row rank m, then A is said to have *full row rank*. A square matrix A for which there exist a matrix B such that $AB = BA = I$, the identity matrix, is called an *invertible* matrix, and B is called the *inverse* of A. A square matrix with full row rank is invertible.

A (*simple*) *graph* G is an ordered set (V, E) where V is a finite set whose elements are called *vertices* and E is a set of unordered pairs of elements of V. The elements of E are called *edges*. We say two edges are *adjacent* if they have a common vertex. A *path* is a sequence of edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}$, where no v_i is repeated. If $v_n = v_1$ it is called a *cycle*. A *tree* is a connected graph with no cycles, and a *forest* is a graph with no cycles. A *supergraph* H of a graph G is a graph on the same vertices as G and each edge of G is also an edge of H. A *spanning subgraph* H of the graph G is a graph on the same vertices as G where each edge of H is also an edge of G. The graph obtained by removing some of the vertices of the graph G and all the edges containing those vertices is called an *induced subgraph* of G.

A (*zero-nonzero*) *pattern* is a matrix $\mathcal{A}$ whose entries are $0, \#$, where $\#$ denotes a nonzero number. We say the matrix A has the (zero-nonzero) pattern $\mathcal{A}$ if $A_{ij} = 0$ whenever $\mathcal{A}_{ij} = 0$, and $A_{ij} \neq 0$ whenever $\mathcal{A}_{ij} = \#$. The pattern $\mathcal{B}$ is said to be a *superpattern* of $\mathcal{A}$ when $\mathcal{A}_{ij} = 0$ if $\mathcal{B}_{ij} = 0$.

The *graph of* an $n \times n$ real symmetric matrix $A = \begin{bmatrix} a_{ij} \end{bmatrix}$ is the (simple) graph G on n vertices $1, 2, \ldots, n$ with edges $\{i, j\}$ if $a_{ij} \neq 0$ for $i \neq j$. The graph of the matrix A is denoted by $\mathcal{G}(A)$, and we denote the set of all real symmetric matrices whose graph is G with $\mathcal{S}(G)$. Note that the graph of a matrix does not depend on the diagonal entries of the matrix. The set of all real symmetric matrices A whose graph is G is denoted by $\mathcal{S}(G)$.

For two sets $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ and $M = \{\mu_1, \mu_2, \ldots, \mu_{n-1}\}$, we say M *interlaces* Λ, if $\lambda_1 \leq \mu_1 \leq \lambda_2 \leq \cdots \leq \lambda_{n-1} \leq \mu_{n-1} \leq \lambda_n$. If all the inequalities above are strict, then the *interlacing is strict.*

Let f be a differentiable function from $\mathbb{R}^m$ to $\mathbb{R}^n$

$$f(x_1, \ldots, x_m) = (f_1(x_1, \ldots, x_m), \ldots, f_n(x_1, \ldots, x_m)),$$

where each f_i is a a real valued multivariate function. The *Jacobian* of f is defined to be the $n \times m$ matrix

$$\mathrm{Jac}(f) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix}.$$

Let A be an $m \times n$ matrix. Assume $\alpha \subseteq \{1, 2, \ldots, m\}$ and $\beta \subseteq \{1, 2, \ldots, n\}$. Then

$A[\alpha; \beta]$ is the matrix obtained from A by keeping the rows indexed by α, and the columns indexed by β;

$A(\alpha; \beta)$ is the matrix obtained from A by deleting the rows indexed by α, and the columns indexed by β;

$A[\alpha; \beta)$ is the matrix obtained from A by keeping the rows indexed by α, and deleting the columns indexed by β; and

$A(\alpha; \beta]$ is the matrix obtained from A by deleting the rows indexed by α, and keeping the columns indexed by β.

We abbreviate $A[\alpha; \alpha]$ to $A[\alpha]$, and $A(\alpha; \alpha)$ to $A(\alpha)$. In case that α or β is a singleton set, we omit the curly brackets. For example, we write $A(1)$ for $A(\{1\})$. Also, in the case that α or β is empty, we may omit them. For example, $A(\,; 1)$ is the submatrix obtained from A by removing the first column. When $\alpha = \beta$ the submatrices $A[\alpha]$ and $A(\alpha)$ are called *principal submatrices* of A. In this case we use the same notation for a graph G, where indices denote vertices. For example $G[X]$ denotes the subgraph of G induced on the vertex set X.

For a vertex v of a graph T we denote the set of all the vertices that are adjacent to v in G by $\mathcal{N}_G(v)$. Each element of $\mathcal{N}_G(v)$ is called a *neighbor* of v in G. In the case that the graph is understood from the context, we may drop the subscript and denote the set of neighbors by $\mathcal{N}(v)$. Consider a tree T and a fixed vertex v of T. The forest obtained from T by removing v from T is denoted by $T(v)$, and the connected component containing the vertex $w \in \mathcal{N}_T(v)$ is denoted by $T_w(v)$. Furthermore, $T_{w'}(v)$ denotes the graph obtained from $T_w(v)$ by removing the vertex w. Similarly, if T is the graph of a matrix A, $A(v)$ denotes the submatrix of A corresponding to indices of vertices in $T(v)$, that is the submatrix obtained by deleting the row and the column v, and $A_w(v)$ denotes the submatrix of A corresponding to the vertices of $T_w(v)$. Furthermore, $A_{w'}(v)$ denotes the matrix obtained from $A_w(v)$ by removing its row w and column w.

A matrix $A \in M_n(\mathbb{R})$ whose graph is a tree T on n vertices $1, 2, \ldots, v, \ldots, n$ is said to have the *Duarte property* with respect to vertex v if

- A is a 1×1 matrix, or
- the eigenvalues of $A(v)$ strictly interlace the eigenvalues of A, and $A_w(v)$ has the Duarte property with respect to w for each $w \in \mathcal{N}_T(v)$.

For two matrices A and B in $M_n(\mathbb{R})$, $A \circ B$ is the *entry-wise product* (also known as *Schur* or *Hadamard product*) of two matrices, that is $(A \circ B)_{ij} = A_{ij}B_{ij}$, and the *commutator*

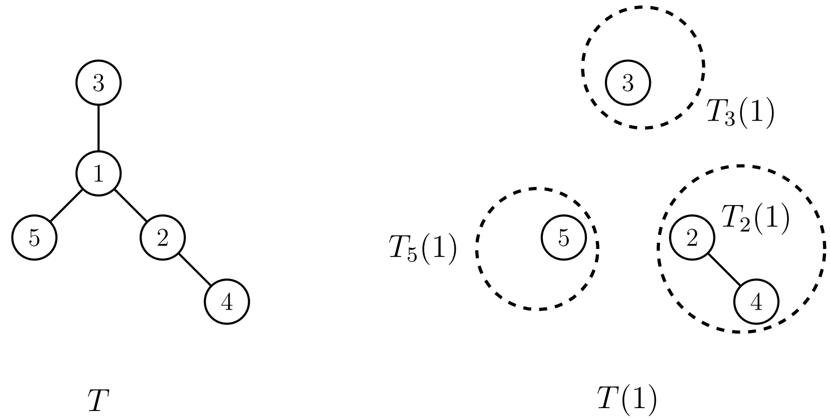


Figure 1.1: A tree and subtrees corresponding to $T(1)$.

of A and B is defined to be $[A, B] = AB - BA$. For two matrices A and B if there is a matrix X such that $AX = XB$, we say A and B *intertwine* (with respect to a matrix X).

As an example, let

$$A = \begin{bmatrix} 30 & -2 & -9 & 0 & 1 \\ -2 & 4 & 0 & -1 & 0 \\ -9 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 4 & 0 \\ 1 & 0 & 0 & 0 & 2 \end{bmatrix}. \tag{1.1}$$

Then

$$A(1) = \begin{bmatrix} 4 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix},$$

and the graph T of A, $T(1)$ and each of the $T_i(1)$'s are illustrated in Figure 1.

The matrices related to each $T_i(1)$ are

$$A_2(1) = \begin{bmatrix} 4 & -1 \\ -1 & 4 \end{bmatrix}, \quad A_3(1) = \begin{bmatrix} -1 \end{bmatrix}, \quad A_5(1) = \begin{bmatrix} 2 \end{bmatrix}.$$

Let f be a function from a subset M of $\mathbb{R}^n$ to $\mathbb{R}^k$. The differential of f at x is the Jacobian matrix of f and is denoted by $d_x f$. Theorem 5.2 of [4] asserts that a subset M of

$\mathbb{R}^n$ is a k dimensional *manifold* if and only if for each point $x \in M$, the following condition is satisfied. There is an open set U containing x, an open set $W \subseteq \mathbb{R}^k$, and a one-to-one differentiable function $f : W \to \mathbb{R}^n$ such that

- $f(W) = M \cap U$,
- $d_x f$ has rank k for each $x \in W$,
- $f^{-1} : f(W) \to W$ is continuous.

We use the above as the working definition of a manifold. Such f is called the *coordinate system* of M around x. If $p \in \mathbb{R}^n$, the set of all pairs $(p, \boldsymbol{v})$ for $\boldsymbol{v} \in \mathbb{R}^n$ is denoted $\mathbb{R}^n_p$, and called the *tangent space* of $\mathbb{R}^n$ at p. Consider a differentiable map $f : \mathbb{R}^n \to \mathbb{R}^m$. This defines a linear transformation $Df(p) : \mathbb{R}^n \to \mathbb{R}^m$, hence we can define another linear transformation $f_{*,p} : \mathbb{R}^n_p \to \mathbb{R}^m_{f(p)}$ defined by

$$f_{*,p}(v) = (Df(p)(v))_{f(p)} \,.$$

Consider a manifold M, and its coordinate system f around $x = f(a)$. Since $d_a f$ has rank k, the linear map $f_{*,a} : \mathbb{R}^k_a \to \mathbb{R}^n_x$ is one-to-one, and $f_{*,a}(\mathbb{R}^k_a)$ is a k dimensional subspace of $\mathbb{R}^n_x$. It can be seen that this subspace is independent of the coordinate system f [4]. This subspace is denoted $T_x(M)$ and is called the *tangent space* of M at x.

Given two manifolds M and N, a differentiable map $f : M \to N$ is called a *diffeomorphism* if it is a bijection and its inverse $f^{-1} : N \to M$ is also differentiable. For a differentiable map $f : M \to N$ a point $x \in M$ is called a *regular point* of f if the linear map $d_x f$ is surjective (has maximal rank, when the dimension of M is less than the dimension of N), otherwise it is called a *critical point*. A point $q \in N$ is a *regular value* of f if all points p in pre-image $f^{-1}(q)$ are regular points.

1.2 Preliminary results

In this section we introduce some preliminary results that will be used later in this dissertation. We first start with the Cauchy interlacing inequalities. Here we only mention the first two orders, similarly one can write the higher order Cauchy inequalities.

Lemma 1.2.1 (Cauchy interlacing inequalities). *Let A be an $n \times n$ real symmetric matrix, with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$, not necessarily distinct. Let B be a principal submatrix of A of size $n-1$ with eigenvalues $\mu_1 \leq \cdots \leq \mu_{n-1}$, and C be an $(n-2) \times (n-2)$ principal submatrix of A with eigenvalues $\tau_1 \leq \cdots \leq \tau_{n-2}$. Then*

$$\lambda_i \leq \mu_i \leq \lambda_{i+1}, \qquad i = 1, \ldots, n-1, \tag{1.2}$$

$$\lambda_i \leq \tau_i \leq \lambda_{i+2}, \qquad i = 1, \ldots, n-2. \tag{1.3}$$

Sketch of proof. Below we provide the sketch of proof for the inequalities (1.2) in the case that $\mu_1 < \mu_2 < \cdots < \mu_{n-1}$. Let

$$A = \left[\begin{array}{c|c} a & \boldsymbol{y}^T \\ \hline \boldsymbol{y} & B \end{array}\right],$$

and let $D = \operatorname{diag}(\mu_1, \ldots, \mu_{n-1})$. There is a unitary matrix U such that $U^T B U = D$. Let

$$V = \left[\begin{array}{c|c} 1 & \mathbf{0}^T \\ \hline \mathbf{0} & U \end{array}\right],$$

where $\mathbf{0}$ is the zero vector of size $n-1$, and assume that $U^T \boldsymbol{y} = \boldsymbol{z}$. Then V^T is a unitary matrix and

$$V^T A V = \left[\begin{array}{c|c} a & \boldsymbol{z}^T \\ \hline \boldsymbol{z} & D \end{array}\right].$$

Let $f(x) = \det(xI - A) = \det(xI - V^T A V)$, and expand f along the first row. It can be seen that

$$f(\mu_i) = -z_i^2 \prod_{j \neq i} (\mu_i - \mu_j).$$

Thus, $f(\mu_i) > 0$, if i is even, and $f(\mu_i) < 0$, if i is odd. Then by intermediate value theorem, and the fact that f is a polynomial with n real roots, one can conclude that $\lambda_1 < \mu_1 < \lambda_2 < \cdots < \mu_{n-1} < \lambda_n$. □

The inequalities (1.2) are called the *first order Cauchy interlacing inequalities*, and the inequalities (1.3) are called the *second order Cauchy interlacing inequalities*. Note that (1.3) are obtained by implying (1.2) twice. If all the inequalities are strict, then we call them the *strict Cauchy interlacing inequalities*. For a detailed proof of Lemma 1.2.1 see [5].

Proposition 1.2.2 (Newton's Identities)**.** *Let* $p(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ *have roots* r_j, $j = 1, 2, \ldots, n$. *Define*

$$s_k \equiv \sum_{j=1}^{n} r_j^k.$$

Then

$$s_k + a_{n-1}s_{k-1} + \cdots + a_0 s_{k-n} = 0, (k > n) \tag{1.4}$$

$$s_k + a_{n-1}s_{k-1} + \cdots + a_{n-k+1}s_1 = -k a_{n-k}, (1 \leq k \leq n) \tag{1.5}$$

For a simple proof or Proposition 1.2.2 using the trace of the companion matrix of $p(x)$ see [6].

Remark 1.2.3. *Newton's identities relate the traces of the powers* A^k *to the coefficients of the characteristic polynomial of* A *with a continuously differentiable function. Let* g *be a function that maps any matrix to the coefficients of its characteristic polynomial and let* f *be a function that maps every matrix to the traces of its powers. Then Newton's identities imply that there is a continuously differentiable function* h *such that* $f = g \circ h$, *and* f *is one-to-one (respectively, onto) in a neighborhood of a point* a *if and only if* g *is one-to-one (respectively, onto) in a neighborhood of* $h(a)$.

Lemma 1.2.4 (Intertwining Lemma)**.** *Let* A *be an* $m \times m$ *matrix,* B *be an* $n \times n$ *matrix, and* X *be an* $m \times n$ *matrix such that* $AX = XB$. *Then the following hold:*

(a) *If* A *and* B *do not have a common eigenvalue, then* $X = O$.

(b) *If $X \neq O$ and A and B share exactly one common eigenvalue, then each nonzero column of X is a generalized eigenvector of A corresponding to the common eigenvalue.*

Proof. Note that the condition $AX = XB$ implies that $p(A)X = Xp(B)$ holds for each polynomial $p(x)$. Let $p(x) = m_B(x)$ be the minimal polynomial of B. Then $m_B(A)X = Xm_B(B) = O$. Hence

$$(A - \mu_1 I)\cdots(A - \mu_{n-1} I)X = O, \tag{1.6}$$

where the μ_i's are the eigenvalues of B. If A and B do not share a common eigenvalue, then each $A - \mu_j I$ is invertible, and it follows that $X = O$.

If A and B share exactly one common eigenvalue, say μ, then each matrix $A - \mu_j I$ with $\mu_j \neq \mu$ is invertible and hence by (1.6), $(A - \mu I)^k X = O$ for some positive integer k. This implies that each nonzero column of X is a generalized eigenvector of A corresponding to the eigenvalue μ. □

The eigenvalues of a matrix are continuous differentiable function of the entries of the matrix in a neighborhood that all the eigenvalues are distinct. Let $\frac{\partial \lambda_i}{\partial a_{jk}} = x_i(j)x_i(k)$ denote the derivative of the i-th smallest eigenvalue of a matrix $A = \left[a_{jk}\right]$ with respect to the entry in the (j,k) position. The following lemma shows how to compute this derivative.

Lemma 1.2.5. *Let A be a real symmetric matrix with distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$, and corresponding unit eigenvectors $x_1, x_2, \ldots, x_n$. Then*

$$\frac{\partial \lambda_i}{\partial a_{jk}} = x_i(j)x_i(k).$$

Proof. Consider $A(t) = A + tE_{jk}$, where $t \in (-\varepsilon, \varepsilon)$ for some small $\varepsilon > 0$. Then $A(t) \to A$, $\lambda_i(t) \to \lambda_i$, and $x_i(t) \to x_i$, when $t \to 0$, and $\dot{A}(0) = E_{jk}$. Furthermore,

$$A(t)x_i(t) = \lambda_i(t)x_i(t).$$

Differentiating both sides with respect to t we get

$$\dot{A}(t)x_i(t) + A(t)\dot{x}_i(t) = \dot{\lambda}_i(t)x_i(t) + \lambda_i(t)\dot{x}_i(t).$$

Set $t = 0$, then

$$E_{jk}x_i + A\dot{x}_i(0) = \dot{\lambda}_i(0)x_i + \lambda_i\dot{x}_i(0).$$

Multiplying both sides by x_i^T from left we get

$$x_i^T E_{jk} x_i + x_i^T A \dot{x}_i(0) = \dot{\lambda}_i(0) x_i^T x_i + \lambda_i x_i^T \dot{x}_i(0).$$

Since A is symmetric

$$x_i(j) x_i(k) + \lambda_i x_i^T \dot{x}_i(0) = \dot{\lambda}_i(0) x_i^T x_i + \lambda_i x_i^T \dot{x}_i(0).$$

But x_i's are unit vectors, hence $x_i^T x_i = 1$. Thus

$$x_i(j) x_i(k) = \dot{\lambda}_i(0).$$

□

For more results related to such derivatives see [7].

1.3 Motivation and Previous Research

> "Vibrations are everywhere, and so too are the eigenvalues associated with them."

Those are the words of B. Parlett in his amazing book "The Symmetric Eigenvalue Problem" [8]. In their book "Inverse Eigenvalue Problems" [9], Chu and Golub mention that

> "Generally speaking, the basic goal of an inverse eigenvalue problem is to reconstruct the physical parameters of a certain system from the knowledge or desire of its dynamical behavior. Since the dynamical behavior often is governed by the underlying natural frequencies and/or normal modes, the spectral constraints are thus imposed. On the other hand, in order that the resulting model is physically realizable, additional structural constraints must also be imposed upon the construction."

In this dissertation we concentrate on a specific case that the underlying physical system is represented by a matrix, and consequently we will discuss inverse eigenvalue problems for matrices. Furthermore, many times the matrix that describes the physical system has a

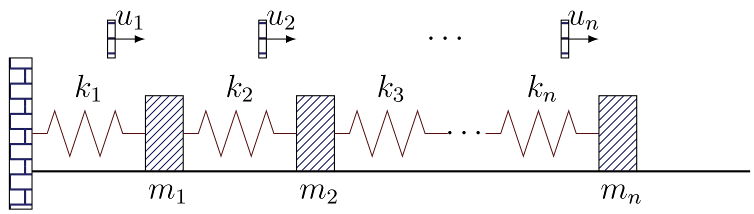


Figure 1.2: A system of masses and springs.

certain structure, for example, its entries are real and the matrix is symmetric. Also, in the cases that we will discuss in this dissertation, we prescribe which entries of the matrix are zero and which ones are nonzero. For detailed discussion of what inverse problems are considered as structured IEP's see [9].

(Structured) inverse eigenvalue problems arise from a variety of applications, such as control design, tomography, particle physics, geophysics, structural analysis, circuit theory, and mechanical system simulation etc. A common theme is that the physical parameters of the underlying system are to be reconstructed from knowledge of its dynamical behavior. "Vibrations depend on natural frequencies and normal modes, stability controls depend on the location of eigenvalues, and so on. As such, the spectral information used to affect the dynamical behavior varies in various ways. If the physical parameters can be, as they often are, described mathematically in the form of a matrix, then we have an IEP. The structure of the matrix is usually inherited from the physical properties of the underlying system" [9].

Here we start with a simple example which arises in mechanical and civil engineering and in robotics. Gladwell in his book "Inverse Problems in Vibration" [10] mentions three systems that are mathematically equivalent:

> Assume that n masses $m_1, m_2, \ldots, m_n$ are connected by a line of springs of stiffnesses $k_1, k_2, \ldots, k_n$, lying on a smooth horizontal surface, and a force $F_i(t)$ is applied to each mass m_i (See Figure 1.2).
>
> Newton's equations of motion for this system are:
>
> $$m_i\ddot{u}_i = F_i + \theta_{i+1} - \theta_i, \text{ for } i = 1, 2, \ldots, n-1, \tag{1.7}$$
>
> $$m_n\ddot{u}_n = F_n - \theta_n \tag{1.8}$$

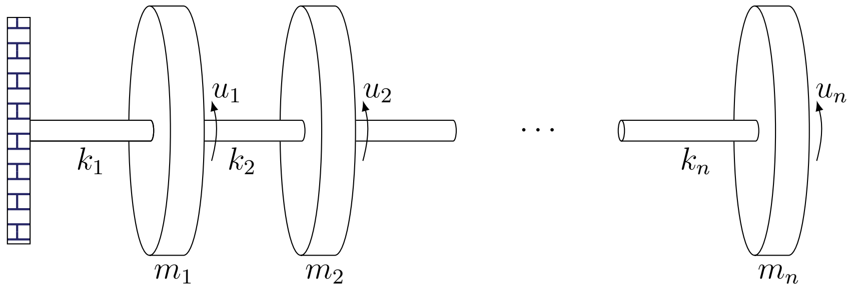


Figure 1.3: A torsionally vibrating system.

where $\cdot$ denotes differentiation with respect to time. Hooke's law implies that the spring forces θ_i are given by:

$$\theta_i = k_i(u_i - u_{i-1}), \text{ for } i = 1, 2, \ldots, n-1. \tag{1.9}$$

If the left hand end is pinned, then

$$u_0 = 0. \tag{1.10}$$

The system shown in Figure 1.2 has considerable engineering importance. It is the simplest possible discrete model for a rod vibrating in longitudinal motion. Here the masses and stiffnesses are obtained by lumping the continuously disturbed mass and stiffness of the rod. Equations (1.7) – (1.10) also describe the torsional vibrations of the system shown in Figure 1.3, provided that the u_i, k_i, and m_i are interpreted as torsional rotation, torsional stiffness, and moments of inertia, respectively. Such a discrete system provides a simple model for the torsional vibrations of a rod with a continuous distribution of inertia and stiffness.

There is a third system which is mathematically equivalent to equations (1.7) – (1.10). This is the transverse motion of the string shown in Figure 1.4 which is pulled taut by a tension T and which is loaded by masses m_i.

In order to express equations (1.7) – (1.9) in matrix form we use (1.9) to obtain

$$m_i\ddot{u}_i = F_i + k_{i+1}u_{i+1} - (k_{i+1} + k_i)u_i + k_iu_{i-1}, \tag{1.11}$$

$$m_n\ddot{u}_n = F_n - k_nu_n + k_nu_{n-1}, \tag{1.12}$$

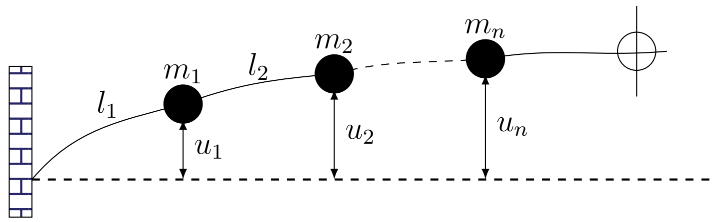


Figure 1.4: A system of masses on a taut string.

which yields

$$\begin{bmatrix} m_1 & & & \\ & m_2 & & \\ & & \ddots & \\ & & & m_n \end{bmatrix} \begin{bmatrix} \ddot{u_1} \\ \ddot{u_2} \\ \vdots \\ \ddot{u_n} \end{bmatrix} + \begin{bmatrix} k_1+k_2 & -k_2 & 0 & \cdots & 0 & 0 \\ -k_2 & k_2+k_3 & -k_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -k_n & k_n \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{bmatrix}. \tag{1.13}$$

This equation may be written as $A\ddot{\boldsymbol{u}}+C\boldsymbol{u} = \boldsymbol{F}$, where the matrices A and C are called respectively the inertia (or mass) and the stiffness matrices of the system. Note that both A and C are symmetric.

When there is no external force involved, that is when $\boldsymbol{F} = \boldsymbol{0}$, then the equations of free vibration may be written as $A\ddot{\boldsymbol{u}} + C\boldsymbol{u} = \boldsymbol{0}$. When $\boldsymbol{u}$ has the form

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \sin(\omega t + \phi),$$

where the constants x_i, frequency ω, and phase angle ϕ are to be determined, the free vibration equations become $\ddot{\boldsymbol{u}} = -\omega^2\boldsymbol{u}$. That is, $(C - \lambda A)\boldsymbol{x} = \boldsymbol{0}$, where $\lambda = \omega^2$. This eigenvalue equation has a solution if and only if $\det(C - \lambda A) = 0$. The frequencies ω_i for each λ_i are called the natural frequencies of the system.

It can be seen that the natural frequencies of a lumped mass system may be obtained as the eigenvalues of a tridiagonal matrix. In this dissertation we are interested in inverse problems that ask about the existence of a vibrating system with given natural frequencies. More generally, one can ask about the existence of a vibrating system where the natural frequencies of the system are given when the right end is free, or when the right hand is not free. Such problems give rise to inverse eigenvalue problems when the eigenvalues of a matrix and the eigenvalues of the matrix after perturbing its last diagonal entry are prescribed. These inverse problems were apparently considered first by Hochstadt [11–15], then by Hald [16, 17], Gray and Wilson [18], and de Boor and Golub [19], and they are closely related to the inverse problems concerning the spectrum of an $n \times n$ matrix and the spectrum of an $(n-1) \times (n-1)$ principal submatrix of it, as illustrated in Section 4.5.

Chu and Golub, in their book “Inverse Eigenvalue Problems” [9], provide a great collection of inverse eigenvalue problems and related literature and previous work. In particular, in [2], the authors concentrate on the structured inverse eigenvalue problems. Some of the important results can be found in [17, 20–30].

More recent results on the specific type of structured inverse eigenvalue problems mentioned here can be found in [31–41], and the most recent work can be found in [42–46].

1.4 Dissertation Overview and Organization

In this dissertation we develop a method called the Jacobian method, which is used to solve three fundamental structured inverse eigenvalue problems. The common strategy is to prove the existence of a solution for trees, then show that these solutions are generic, and then use the Jacobian method to extend it to connected graphs.

In Chapter 2, the λ-structured inverse eigenvalue problem (SIEP) is solved (see Theorem 2.3.4). We also give a geometric interpretation of what is happening in the proof of this theorem, by describing some manifolds, showing they are smooth at some point and they intersect transversally at that point, and finally we use a version of the Implicit Function Theorem (IFT) to prove Theorem 2.3.4 geometrically.

In Chapter 3, a result of A.L. Duarte which gives an answer to the λ-μ-SIEP for trees is mentioned and proved (see Theorem 3.1.1). Then we use the Jacobian method to show there exist a solution for the λ-μ-SIEP for connected graphs (see Theorem 3.3.1).

In Chapter 4, the existence of a solution for the λ-τ-SIEP for trees in two cases are stated and proved (see Theorems 4.2.1 and 4.4.2). Then we use the Jacobian method to show the existence of a solution for the λ-τ-SIEP for connected graphs (see Theorems 4.3.4 and 4.4.4). Furthermore, in Chapter 4, we use the mentioned results and solve problems related to perturbing one or two diagonal entries of a matrix so that the eigenvalues of the new matrix change as prescribed (see Theorems 4.5.5 and 4.5.6).

In Chapter 5, it is first shown that for any given tree and a set of distinct real numbers, there is a matrix with that tree and spectrum such that none of the eigenvectors of the matrix have a zero entry (see Theorem 5.1.8). Then we extend that result to any connected graph (see Theorem 5.1.9). In the proof, both results from the λ-μ-SIEP and the λ-τ-SIEP are used.

In Chapter 6, a series of problems that could be solved using the Jacobian method are proposed and discussed.

Throughout this dissertation we use interesting mathematical ideas, tools, and techniques, and we prove combinatorial results and algebraic results that individually are interesting, important, and might be used elsewhere, such as the properties of the λ-τ sequence of a graph, techniques used for proving the nonsingularity of some matrices, and results related to intertwining matrices etc.

The above mentioned theorems suggest algorithms to construct (sometimes approximately) a matrix with the prescribed spectral data and graph. We provide the SAGE code developed by author for the λ-SIEP for graphs, λ-μ-SIEP for trees, and λ-τ-SIEP for trees, in Appendix A, as well as some sample inputs and outputs.

Chapter 2

The Jacobian Method and the λ-structured Inverse Eigenvalue Problem

This chapter is devoted to describing a method that we call the *Jacobian* method. As Bryan Shader describes it, if you find a 'generic' needle in a haystack, then you can find more needles in nearby haystacks.

One of the most common themes in mathematical research problems is to find a mathematical object which satisfies certain properties, and many times one seeks to find all such objects. In the Jacobian method one tries to find a solution to the concerned problem such that the solution is 'generic'. That is, small perturbations of some parameters of the solution can be adjusted by adjusting other parameters so that the defining properties of the solutions do not change, hence it results in a new solution to that problem, or to a slightly different problem. This means the solution that we have started with, is robust. While the Jacobian method can be used in various mathematical settings, in this dissertation we limit ourselves to a formulation of this method for some structured inverse eigenvalue problems (SIEP's), and give a geometric interpretation of the conditions to be satisfied.

In Section 2.1 we use the Implicit Function Theorem to provide some simple examples. In Section 2.2 we illustrate the main ideas of the Jacobian method by solving the λ-structured inverse eigenvalue problem. In Section 2.3 we provide rigorous details for the Jacobian method used to solve the λ-SIEP. Finally in Section 2.4 we give a geometric interpretation of this method, and an idea on how to solve the λ-SIEP geometrically.

2.1 The Implicit Function Theorem

The Jacobian method is nothing but a clever application of the Implicit Function Theorem (IFT) in a specific settings. So, let us take a look at a version of the IFT that best fits our needs. There are various formulations of the IFT which are slightly different from each other. Many of these various formulations are sought after in Krantz's beautiful book "The Implicit Function Theorem" [47]. Below is a statement of the IFT which we use in this research.

Theorem 2.1.1 (Implicit Function Theorem). *Let $F : \mathbb{R}^{s+r} \to \mathbb{R}^s$ be a continuously differentiable function on an open subset U of $\mathbb{R}^{s+r}$ defined by*

$$F(\boldsymbol{x}, \boldsymbol{y}) = (F_1(\boldsymbol{x}, \boldsymbol{y}), F_2(\boldsymbol{x}, \boldsymbol{y}), \ldots, F_s(\boldsymbol{x}, \boldsymbol{y})),$$

where $\boldsymbol{x} = (x_1, \ldots, x_s) \in \mathbb{R}^s$, $\boldsymbol{y} = (y_1, \ldots, y_r) \in \mathbb{R}^r$, and F_i's are real valued multivariate functions. Let $(\boldsymbol{a}, \boldsymbol{b})$ be an element of U with $\boldsymbol{a} \in \mathbb{R}^s$ and $\boldsymbol{b} \in \mathbb{R}^r$, and $\boldsymbol{c}$ be an element of $\mathbb{R}^s$ such that $F(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{c}$. If

$$\mathrm{Jac}_x(F)\Big|_{(\boldsymbol{a},\boldsymbol{b})} = \left[\frac{\partial F_i}{\partial x_j}\Big|_{(\boldsymbol{a},\boldsymbol{b})}\right]_{s\times s}$$

is nonsingular, then there exist an open neighborhood V of $\boldsymbol{a}$ and an open neighborhood W of $\boldsymbol{b}$ such that $V \times W \subseteq U$ such that for each $\boldsymbol{y} \in W$ there is an $\boldsymbol{x} \in V$ with $F(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{c}$. Furthermore, for any $(\bar{\boldsymbol{a}}, \bar{\boldsymbol{b}}) \in V \times W$ such that $F(\bar{\boldsymbol{a}}, \bar{\boldsymbol{b}}) = \boldsymbol{c}$, $\mathrm{Jac}(F)\Big|_{(\bar{\boldsymbol{a}},\bar{\boldsymbol{b}})}$ is also nonsingular.

Below are some very simple examples illustrating how the Implicit Function Theorem will be used in this dissertation.

Example 2.1.2. *Let $f(x, y) = x^2 + (y + 1)^2 - 4$, and consider the circle $f = 0$ in the plane. We want to find a point on the circle such that the coordinates of the point are both nonzero. We first find the point $(a, 0) = (\sqrt{3}, 0)$ on the circle. The tangent line to the circle at this point is not horizontal. Let*

$$\mathrm{Jac}(f) = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 2x & 2y + 2 \end{bmatrix}.$$

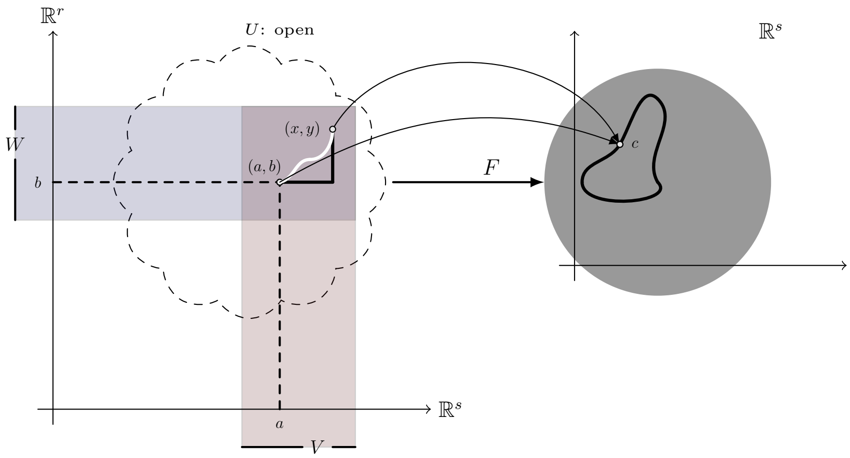


Figure 2.1: The whole white curve (r dimensional surface) on the left is mapped by F to the point c on the right.

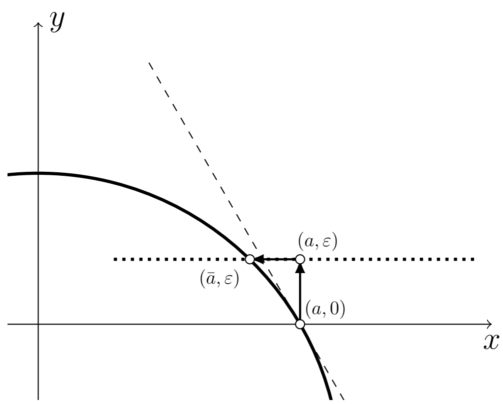


Figure 2.2: Perturbing the zero entry of a point on the circle and adjusting the nonzero entry to get back on the circle.

Then

$$\mathrm{Jac}(f)\Big|_{(\sqrt{3},0)} \left[\begin{array}{cc} 2\sqrt{3} & 2 \end{array}\right],$$

has full row rank. In particular, the submatrix of $\mathrm{Jac}(f)\Big|_{(\sqrt{3},0)}$ *corresponding to the variable* x *is nonsingular. Recall that we want to find a point on the circle such that none of its coordinates are zero. We simply perturb the zero entry of the point by a small* ε*. The point* (a, ε) *is not necessarily on the circle, but since* $\mathrm{Jac}_x(f)\Big|_{(\sqrt{3},0)}[1]$ *is nonsingular, the Implicit Function Theorem guarantees the existence of an* $\bar{a}$ *close to* a *such that* $(\bar{a}, \varepsilon)$ *is back on the circle. In other words* $f(\bar{a}, \varepsilon) = 0$.

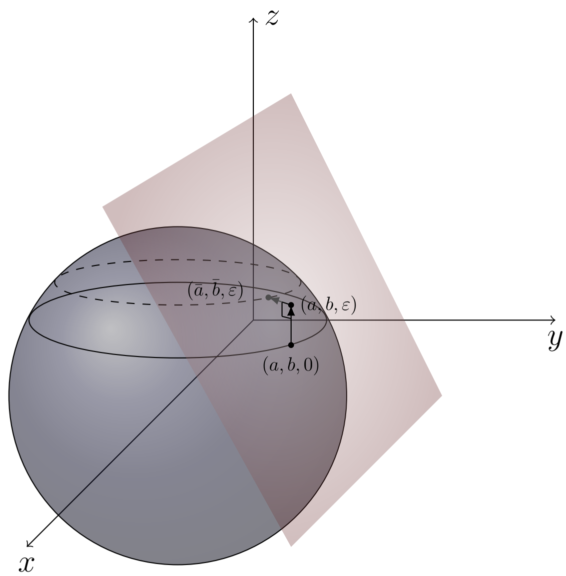


Figure 2.3: Perturbing the zero entry of the point $(a, b, 0)$ on the sphere and adjusting the nonzero entires to get back on the sphere.

Example 2.1.3. *Let* $f(x, y, z) = x^2 + (y+1)^2(z+1)^2 - 5$. *We want to find a point on the sphere* $f = 0$ *so that all of its coordinates are nonzero. First find a point where the third coordinate is zero. This can be done by setting* $z = 0$ *in* $f(x, y, z) = 0$ *and finding a point on the circle* $x^2 + (y+1)^2 - 4 = 0$ *using the method explained in Example 2.1.2. If the point is chosen carefully it can be shown that the tangent plane to the sphere at that point is not horizontal (it is not perpendicular to the z-axis) — in this case any point* $(a, b, 0)$ *on the circle works. Now, it is enough to perturb the zero entry to some small enough* ε*, and adjust* a *and* b *accordingly to get back on the sphere. The point* $(\bar{a}, \bar{b}, \varepsilon)$ *on the sphere has all nonzero entries.*

2.2 The λ-Structured Inverse Eigenvalue Problem

Let us consider a simple structured inverse eigenvalue problem.

Problem 1. ***The λ-SIEP for graphs:*** *A set of distinct real numbers* $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ *and a graph* G *on* n *vertices* $1, 2, \ldots, n$ *are given. Find a real symmetric matrix* A *such that* $\mathcal{G}(A) = G$*, and* $\sigma(A) = \Lambda$.

If we do not take the graph of A into account, then $\mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ is a solution to this problem. Wayne Barrett et al. [48, 49] showed that for certain graphs the matrix A can be constructed by considering some orthogonal similarities of $\mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. For example, choosing the orthogonal symmetries carefully, in order to not make any entry of the final matrix zero, one can achieve the complete graph on n vertices for a certain prescribed Λ. Furthermore, they characterize all graphs and Λ's that can be realized by a real symmetric $n \times n$ matrix, using this method for $n \leq 4$.

Here we provide a complete solution for the λ-SIEP for graphs.

Theorem 2.2.1. *Let $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ be n distinct real numbers. Given a graph G on n vertices $1, 2, \ldots, n$ there is a real symmetric matrix A such that $\sigma(A) = \Lambda$ and $\mathcal{G}(A) = G$.*

Proof. Let $A = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, it is clear that $\sigma(A) = \Lambda$. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ where $x_1, \ldots, x_n$, and $y_1, \ldots, y_m$ are independent real variables, and m is the number of edges of G. Define the $n \times n$ symmetric matrix $M(x, y)$, denoted by M for short, where $M_{i,i} = x_i$ for all $i = 1, \ldots, n$, and $M_{i_k,j_k} = y_k$ if $\{i_k, j_k\}$ is an edge of G. We want to find $(\bar{\boldsymbol{a}}, \boldsymbol{\varepsilon}) \in \mathbb{R}^n \times \mathbb{R}^m$ such that none of the entries of $\boldsymbol{\varepsilon}$ are zero, and $\sigma(\bar{A}) = \Lambda$, where $\bar{A} = M(\bar{\boldsymbol{a}}, \boldsymbol{\varepsilon})$. Note that if none of the entries of $\boldsymbol{\varepsilon}$ are zero, then $\mathcal{G}(\bar{A}) = G$.

Define a function g from the set of $n \times n$ real symmetric matrices with distinct eigenvalues to $\mathbb{R}^n$, and let g map each matrix A in the domain to the sorted n-tuple of its eigenvalues $(\lambda_1(A), \ldots, \lambda_n(A))$, where $\lambda_i(A) < \lambda_{i+1}(A)$, for $i = 1, \ldots, n-1$. Thus

$$g(\lambda_1, \ldots, \lambda_n, 0, \ldots, 0) = (\lambda_1, \ldots, \lambda_n). \tag{2.1}$$

Lemma 1.2.5 implies that

$$\mathrm{Jac}(g)\Big|_{(\lambda_1, \ldots, \lambda_n, 0, \ldots, 0)} = \left[\begin{array}{c|c} I & O \end{array}\right].$$

So, it has full row rank. Then by Implicit Function Theorem there are open sets $U \in \mathbb{R}^n$ and $V \in \mathbb{R}^m$, such that $(\lambda_1, \ldots, \lambda_n) \in U$ and $(0, \ldots, 0) \in V$, and for any $(\varepsilon_1, \ldots, \varepsilon_m) \in V$, there is a $(\overline{\lambda_1}, \ldots, \overline{\lambda_n}) \in U$ close to $(\lambda_1, \ldots, \lambda_n)$, such that $g(\overline{\lambda_1}, \ldots, \overline{\lambda_n}, \varepsilon_1, \ldots, \varepsilon_m) = (\lambda_1, \ldots, \lambda_n)$. Since V is an open neighborhood of $(0, \ldots, 0) \in \mathbb{R}^m$, one can choose all $\varepsilon_i \neq 0$. Let $\overline{A} = M(\overline{\lambda_1}, \ldots, \overline{\lambda_n}, \varepsilon_1, \ldots, \varepsilon_m)$. Then $\sigma(\overline{A}) = \Lambda$ and $\mathcal{G}(A) = G$. $\square$

2.3 Difficulties and Solutions

The solution of the λ-SIEP given in Theorem 2.2.1 illustrates the essential ideas of the Jacobian method. As we try to apply the Jacobian method in more complex settings, we encounter difficulties in computing the Jacobian and showing the Jacobian matrix is nonsingular. We overcome the former issue by considering a more amenable function F which is closely related to the function g defind by 2.1. We overcome the latter issue by showing the rows of the Jacobian matrix are linearly independent, rather than showing its determinant is nonzero. In this section, we illustrate the ways we overcome the two issues in the setting of the simple λ-SIEP.

In order to find a matrix whose graph is G, we first find a matrix for a subgraph H of G which has the same number of vertices but usually has fewer edges. Then we perturb the zero entries of A corresponding to the edges of G not in H, to make them nonzero. We call this new matrix $\tilde{A}$. Then $\mathcal{G}(\tilde{A}) = G$, but the eigenvalues of $\tilde{A}$ are different from those of A. Since λ_i's are distinct, the eigenvalues are continuous (and differentiable) functions of entries of the matrix [50]. If the perturbations are small, then the eigenvalues of $\tilde{A}$ are close to the eigenvalues of A. Now, we adjust the diagonal entries of $\tilde{A}$ so that the eigenvalues of the new matrix $\bar{A}$ are λ_i's. This is where the Implicit Function Theorem is used to show this perturbation and adjustment is possible, provided that we start with a 'generic' matrix A.

We can work with different functions, but ideally we want a function g that maps a matrix to its eigenvalues. While g does the job for the λ-SIEP, it is hard to work with the derivatives of g in general. In order to make the Jacobian simple, we define a function F that maps each matrix whose graph is G to the coefficients of its characteristic polynomial, that is,

$$F(\boldsymbol{x}, \boldsymbol{y}) = (c_0(\boldsymbol{x}, \boldsymbol{y}), c_1(\boldsymbol{x}, \boldsymbol{y}), \ldots, c_{n-1}(\boldsymbol{x}, \boldsymbol{y})),$$

where $c_i(\boldsymbol{x}, \boldsymbol{y})$ is the coefficient of x^i in the characteristic polynomial of $M(\boldsymbol{x}, \boldsymbol{y})$. This function is well-defined over the set of square matrices. We restrict our domain to the set of real symmetric matrices whose graph is a subgraph of G, for now.

Let $(\boldsymbol{a}, \mathbf{0}) \in \mathbb{R}^n \times \mathbb{R}^m$ be an assignment of $\boldsymbol{x}$ and $\boldsymbol{y}$ such that $M(\boldsymbol{a}, \mathbf{0}) = A$. In order to use the Implicit Function Theorem, it suffices to show that the Jacobian of F evaluated at $(\boldsymbol{a}, \mathbf{0})$, denoted by $\operatorname{Jac}(F)\big|_{(A,\mathbf{0})}$, has full row rank. Columns of this Jacobian matrix evaluated at $(\boldsymbol{a}, \mathbf{0})$ correspond to the derivatives of F with respect to variables x_i's and y_j's. Let $\operatorname{Jac}_x(F)\big|_A$ denote the matrix obtained from the above Jacobian matrix by deleting the columns corresponding to the derivatives of F with respect to y_j's. We prove that $\operatorname{Jac}_x(F)\big|_{(A,\mathbf{0})}$ has full row rank. Consequently $\operatorname{Jac}(F)\big|_{(A,\mathbf{0})}$ has full row rank. Then the Implicit function Theorem guarantees the existence of $(\bar{\boldsymbol{a}}, \boldsymbol{\varepsilon})$ such that $\sigma(A) = \Lambda$ and $\mathcal{G}(A) = G$.

One way to approach this problem is to differentiate the functions above and take the determinant of the Jacobian, after evaluating it at $(A, \mathbf{0})$, in order to show that it is nonzero. It is hard to work with the derivatives of the coefficients of the characteristic polynomial of a matrix with respect to the entries of the matrix. To illustrate this, let $M(x_1, \ldots, x_n, 0, \ldots, 0) = \operatorname{diag}(x_1, \ldots, x_n)$, then the characteristic polynomial of A is

$$C_A(x) = x^n - \left(\sum_{i=1}^{n} x_i\right) x^{n-1} + \left(\sum_{i<j} x_i x_j\right) x^{n-2} - \cdots + (-1)^n x_1 \cdots x_n.$$

Differentiating F with respect to x_i's and evaluating them at the matrix $A = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ we get:

$$\operatorname{Jac}(F)\big|_{(A,\mathbf{0})} = \begin{bmatrix} (-1)^n \prod_{i\neq 1} \lambda_i & \cdots & (-1)^n \prod_{i\neq n} \lambda_i \\ (-1)^{n-1} \sum_{i\neq 1} \left(\prod_{j\neq i,1} \lambda_j\right) & \cdots & (-1)^{n-1} \sum_{i\neq n} \left(\prod_{j\neq i,n} \lambda_j\right) \\ \vdots & \ddots & \vdots \\ -\sum_{i\neq 1} \lambda_i & \cdots & -\sum_{i\neq n} \lambda_i \end{bmatrix}.$$

So, we take a different approach here. First, let M be as before, except each diagonal entry $M_{ii} = 2x_i$, for simplicity of partial derivatives. Let $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{2n-1}$ be the polynomial map defined by

$$f(\boldsymbol{x}, \boldsymbol{y}) = \left(\frac{\operatorname{tr} M(\boldsymbol{x}, \boldsymbol{y})}{2}, \frac{\operatorname{tr} M^2(\boldsymbol{x}, \boldsymbol{y})}{4}, \ldots, \frac{\operatorname{tr} M^n(\boldsymbol{x}, \boldsymbol{y})}{2n}\right). \tag{2.2}$$

By Newton's identities (1.2.2), there is an infinitely differentiable, invertible $h : \mathbb{R}^{n+m} \to \mathbb{R}^{n+m}$ such that $F(\boldsymbol{x}, \boldsymbol{y}) \circ h(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}, \boldsymbol{y})$. Thus, the Jacobian matrix of f at a point $(\boldsymbol{x}, \boldsymbol{y})$

is nonsingular if and only if the Jacobian matrix of $F(\boldsymbol{x}, \boldsymbol{y})$ at $h(\boldsymbol{x}, \boldsymbol{y})$ is nonsingular. Next, we give a closed formula for the Jacobian matrix of the map $f(\boldsymbol{x}, \mathbf{0})$.

Lemma 2.3.1. *Let (i,i) be a diagonal position of A with corresponding variable x_i in M. Then*

$$\frac{\partial}{\partial x_i}\left(\operatorname{tr} M^k(x,0)\right) = 2k\,\left(M^{k-1}(x,0)\right)_{i,i}.$$

Proof. First note that

$$\frac{\partial}{\partial x_i} M = 2E_{ii}.$$

Thus,

$$\begin{aligned}
\frac{\partial}{\partial x_i}\left(\operatorname{tr}(M^k)\right) &= \sum_{\ell=0}^{k-1} \operatorname{tr}\left(M^{\ell} \cdot \frac{\partial}{\partial x_i} M \cdot M^{k-\ell-1}\right) && \text{(by the chain rule)}\\
&= \sum_{\ell=0}^{k-1} \operatorname{tr}\left(M^{k-1} \cdot \frac{\partial}{\partial x_i} M\right) && \text{(since } \operatorname{tr}(AB) = \operatorname{tr}(BA) \text{ for any } A \text{ and } B\text{)}\\
&= k \operatorname{tr}\left(M^{k-1}(2E_{ii})\right)\\
&= 2k\left(M^{k-1}\right)_{i,i}.
\end{aligned}$$

$\square$

Corollary 2.3.2. *Let f be defined by (2.2). Then*

$$\operatorname{Jac}_x(f)\Big|_{(A,\mathbf{0})} = \begin{bmatrix} I_{11} & I_{22} & \cdots & I_{nn} \\ A_{11} & A_{22} & \cdots & A_{nn} \\ \vdots & \vdots & \ddots & \vdots \\ A_{11}^{n-1} & A_{22}^{n-1} & \cdots & A_{nn}^{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_n \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \cdots & \lambda_n^{n-1} \end{bmatrix}.$$

The right hand side in Corollary 2.3.2 is a *Vandermonde* matrix. There are many proofs which show that the Vandermonde matrix is nonsingular when the λ_i's are distinct. Here we prove this by showing the rows of the matrix are linearly independent.

Lemma 2.3.3. *Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be n distinct numbers, and*

$$V = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_n \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \cdots & \lambda_n^{n-1} \end{bmatrix}.$$

Then V is invertible.

Proof. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ and $p(x) = \alpha_1 x^{n-1} + \alpha_2 x^{n-2} + \cdots + \alpha_{n-1} x + \alpha_n$.

Note that $\boldsymbol{\alpha}^T V = \mathbf{0}^T$ if and only if $p(\lambda_j) = 0$ for $j = 1, 2, \ldots, n$. Since $\deg(p) < n$ and λ_i's are distinct, $p(\lambda_j) = 0$ for all j if and only if $p(x)$ is the zero polynomial. That is, if $\alpha_j = 0$ for $j = 1, 2, \ldots, n$. □

Now we are ready to provide another proof for the existence of a solution to the λ-SIEP.

Theorem 2.3.4. *Let $\lambda_1, \ldots, \lambda_n$ be n distinct real numbers, and let G be a fixed graph on n vertices $1, 2, \ldots, n$. There is a real symmetric matrix A whose graph is G such that $\lambda_1 \ldots, \lambda_n$ are eigenvalues of A.*

Proof. Let $A = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, and $\boldsymbol{a}$ be an assignment of variables x_i's corresponding to the diagonal entries of A. Let U be an open neighborhood of $(\boldsymbol{a}, \mathbf{0})$ such that the first n entries of each vector in U are nonzero. By the Implicit Function Theorem there is an open neighborhood $V \subseteq \mathbb{R}^n$ of $\boldsymbol{a}$ and an open neighborhood $W \subseteq \mathbb{R}^m$ of $\mathbf{0}$ such that $V \times W \subseteq U$ and for each $\boldsymbol{y} \in W$ there is an $\boldsymbol{x} \in V$ such that

$$f(\boldsymbol{x}, \boldsymbol{y}) = \left(\frac{\operatorname{tr}(A)}{2}, \ldots, \frac{\operatorname{tr}(A^n)}{2n}\right). \tag{2.3}$$

Take $\boldsymbol{y}$ to be a vector in W with no zero entries. Then the $(\boldsymbol{x}, \boldsymbol{y})$ satisfying (2.3) corresponds to a matrix $M(\boldsymbol{x}, \boldsymbol{y}) = \bar{A} \in S(G)$ such that $\sigma(A) = \Lambda$.

□

Example 2.3.5. *Let K_5 be the complete graph on 5 vertices. We want to realize a matrix whose graph is K_5 and has eigenvalues $1, 2, 3, 4$, and 5. We start with the diagonal matrix $A_0 = \operatorname{diag}(1, 2, 3, 4, 5)$. The Jacobian of the function f defined by (2.3) evaluated at A_0 is nonsingular. So, the Implicit Function Theorem implies that*

$$f(x_1, x_2, \ldots, x_5, \varepsilon, \varepsilon, \ldots, \varepsilon) = \left(\frac{15}{2}, \frac{55}{4}, \frac{225}{6}, \frac{979}{8}, \frac{4425}{10}\right)$$

has a solution for sufficiently small ε. Equivalently,

$$F(x_1, x_2, \ldots, x_5, \varepsilon, \varepsilon, \ldots, \varepsilon) = (-120, 274, -225, 85, -15)$$

has a solution, for sufficiently small ε. For example we find an approximate solution with $\varepsilon = 0.1$ using Newton's method which after 100 iterations gives

$$A_{100} \approx \begin{bmatrix} 1.01822451 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 2.00957906 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 3.00248021 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 3.99393101 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 4.97578522 \end{bmatrix}.$$

It is easy to check that A_{100} has eigenvalues approximately 5, 4, 3, 2, 1, and its graph is K_5.

Remark 2.3.6. *For a given set of n distinct real numbers $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ and a graph G on n vertices the above discussion implies the existence of a matrix of the form $A = D + \varepsilon B$, where D is a diagonal matrix, ε is a positive real number, and B is the adjacency matrix of G, such that $\sigma(A) = \Lambda$. It is clear that $\mathcal{G}(A) = G$.*

Remark 2.3.7. *In Example 2.3.5 it can be seen that if we start with all positive eigenvalues, and choose all $\varepsilon > 0$ small enough, it is guaranteed that the final matrix has all of its entries to be positive, yielding a solution to the positive SIEP below.*

Problem 2. ***The positive SIEP with positive eigenvalues:*** *A set of distinct positive real numbers $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ is given. Find a positive symmetric matrix A such that $\sigma(A) = \Lambda$.*

2.4 Geometric Interpretation: Genericity and Transversality

In the settings of the λ-SIEP a *generic solution* is a matrix such that the Jacobian of the function f defined by (2.2) is nonsingular. In the Jacobian method we are always looking for a generic solution which has a certain zero-nonzero pattern, and a certain spectral property. For example, in the λ-SIEP we want to start with a matrix whose graph is a subgraph of all graphs on n vertices, namely the empty graph on n vertices, and its spectrum is $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$, where λ_i's are distinct real numbers. That is, we are finding the intersection of two sets

$$P = \{B \in M_n(\mathbb{R}) \,|\, B \text{ is a diagonal matrix}\},$$

and

$$S = \{B \in \mathrm{Sym}_n(\mathbb{R}) \,|\, \sigma(B) = \Lambda\}.$$

One point of intersection is $A = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. Note that P is an n dimensional manifold, and S (in a sufficiently small neighborhood of A) is also a manifold. To see this, let us first mention the following proposition, which is a modified version of Theorem 5.1 of [4].

Proposition 2.4.1 (Regular Level Set Theorem)**.** *Given a smooth map $f : \mathbb{R}^m \to \mathbb{R}^n$, let $y \in \mathbb{R}^n$ be a regular value. Then a nonempty level set $f^{-1}(y)$ is a smooth $(m-n)$ dimensional manifold.*

Proof. Let $x \in f^{-1}(y)$. Define $K = \ker(d_x f)$. Then K is the tangent space to $f^{-1}(y)$ at x. Let π be the projection map $\pi : \mathbb{R}^m \to K$. Define $F : \mathbb{R}^m \to \mathbb{R}^n \times K$ by

$$F(x) = (f(x), \pi(x))\,.$$

It is clear that F is a map of manifolds of the same dimension, and $d_x F = (d_x f, \pi)$. Then $d_x F$ is nondegenerate, and F is locally a diffeomorphism. Hence $F\left(f^{-1}(y)\right) = y \times K$ is smooth manifold, thus $f^{-1}(y)$ is a smooth manifold. □

Define a function $\phi : P \to \mathbb{R}^n$ by

$$\mathrm{diag}(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_n).$$

The map ϕ is a diffeomorphism, thus by definition P is a smooth manifold. Now, define $\psi : S \to \mathbb{R}^n$ by

$$A \mapsto (c_0, c_1, \ldots, c_{n-1}),$$

where the c_i's are the nonleading coefficients of the characteristic polynomial of A. Using Proposition 2.4.1 it can be seen that S (in a sufficiently small neighborhood of A) is a

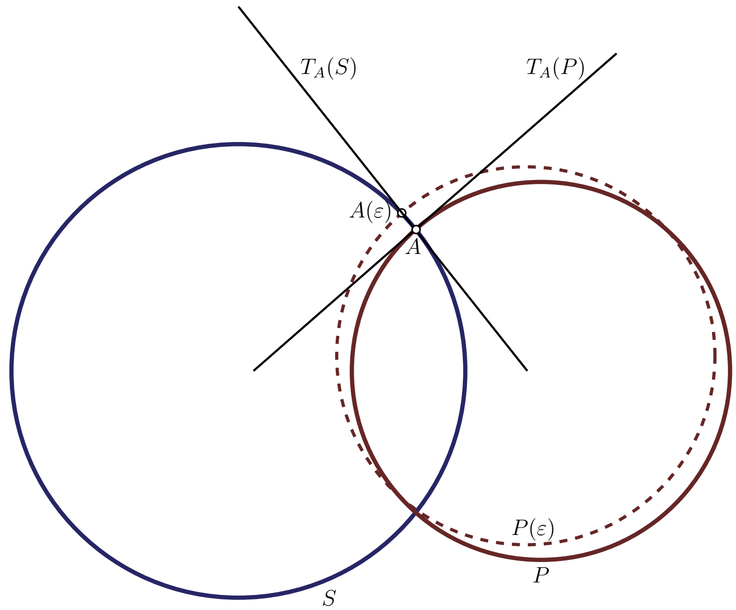


Figure 2.4: Two manifolds P and S intersect transvesally at A.

manifold. It suffices to let y to be the vector of the nonleading coefficients of the characteristic polynomial of A, a regular value of ψ, since the λ_i's are distinct.

In differential geometry the concept of 'generic intersection' is explained by the notion of 'transversality'. "Two submanifolds of a finite dimensional smooth manifold intersect transversally at some point if their tangent spaces at that point together span the tangent space of the ambient manifold" [51]. If we let the ambient smooth manifold to be $\mathbb{R}^N$ for some positive integer N, the transverse intersection of two manifolds P and S means their tangent spaces span the whole $\mathbb{R}^N$. Equivalently, P and S intersect transversally at a point of intersection A, if their normal spaces $P^\perp$ and $S^\perp$ at A are 'independent', that is, no matter how we select a normal vector of P and S at A, they are linearly independent [51].

"Transversal intersections are nice because near them the manifolds behave like subspaces" [51]. The following is a version of the Implicit Function Theorem, a specific version of the Lemma 2.1 of [51]. A smooth family of manifolds $M(t)$ in $\mathbb{R}^N$ is defined by a smooth function $f : U \times (-1, 1) \to \mathbb{R}^N$, where U is an open set in $\mathbb{R}^N$, and for each $t \in (-1, 1)$, the function $f(\cdot, t)$ is a diffeomorphism between U and the manifold $M(t)$.

Lemma 2.4.2. *Let $P(t)$ and $S(t)$ be smooth families of manifolds in $\mathbb{R}^N$, for some positive* *integer N, and assume that $P(0)$ and $S(0)$ intersect transversally at A. Then there exists a neighborhood $W \subseteq \mathbb{R}^2$ of the origin, such that for each $\varepsilon \in W$, the manifolds $P(\varepsilon_1)$ and $S(\varepsilon_2)$ intersect transversally at a point $A(\varepsilon)$, so that $A(0) = A$ and $A(\varepsilon)$ depends continuously on ε.*

Recall that in in our setting

$$P = \{B \in \mathrm{Sym}_n(\mathbb{R}) \,|\, B \text{ is a diagonal matrix}\},$$

and

$$S = \{B \in \mathrm{Sym}_n(\mathbb{R}) \,|\, \sigma(B) = \Lambda\},$$

and in the Jacobian method the aim is to show P and S intersect transversally at $A = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. In order to show that $P(0) = P$ and $S(0) = S$ intersect transversally at A, define a path in S as follows:

$$A(t) = Q(t)\, A\, Q(t)^T,$$

for a family of orthogonal matrices $Q(t)$, such that $Q(0) = I$, the identity matrix. Then $A(0) = A$.

$$\dot{A}(t) = \dot{Q}(t)\, A\, Q(t) + Q(t)\, A\, \dot{Q}(t).$$

Hence

$$\dot{A}(0) = \dot{Q}(0)\, A + A\, \dot{Q}(0)^T.$$

It is not hard to show that the set of all skew-symmetric matrices, that is, $\{B : B^T = -B\}$, is the tangent space to the orthogonal matrices at I [51]. Hence, the tangent space to S at A is

$$T_A(S) = \{BA - AB : B^T = -B\}.$$

Recall that $A = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. Let $B = \left[\, B_{ij} \,\right]$. Then

$$BA - AB = \begin{bmatrix} (\lambda_1 - \lambda_1)B_{11} & (\lambda_2 - \lambda_1)B_{12} & \cdots & (\lambda_n - \lambda_1)B_{1n} \\ (\lambda_1 - \lambda_2)B_{21} & (\lambda_2 - \lambda_2)B_{22} & \cdots & (\lambda_n - \lambda_2)B_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ (\lambda_1 - \lambda_n)B_{n1} & (\lambda_1 - \lambda_1)B_{n2} & \cdots & (\lambda_n - \lambda_n)B_{nn} \end{bmatrix}.$$

Note that the matrix $BA - AB$ is a symmetric matrix with all diagonal entries equal to zero since B is skew-symmetric. Let $W = \begin{bmatrix} W_{ij} \end{bmatrix}$ be a real symmetric matrix with $W_{ii} = 0$ for all i. Also let $B_{ij} = \frac{W_{ij}}{\lambda_i - \lambda_j}$ for $i \neq j$. Then $W = BA - AB$. That is, any real symmetric matrix W with all the diagonal entries equal to zero can be written as $W = BA - AB$ for some skew-symmetric matrix B. Hence $T_A(S)$ is the set of all real symmetric matrices with zero diagonal. So dimension of $T_A(S)$ is $\frac{n(n-1)}{2}$.

On the other hand, in order to find the tangent space to P at A note that the set of diagonal matrices is a subspace, hence its tangent space at any point is itself. That is

$$T_A(P) = \{D | D \text{ is a diagonal matrix}\}.$$

Hence $T_A(P)$ is an n dimensional space.

Finally, note that $T_A(P) \cap T_A(S) = \emptyset$, and thus $\mathrm{Sym}_n(\mathbb{R}) = T_A(P) \oplus T_A(S)$, that is P and S intersect transversally at A. By perturbing A to $\widehat{A}$ in the directions orthogonal to P we can realize any supergraph of G by $\widehat{A} \in P(\varepsilon)$, and then Lemma 2.4.2 guarantees that $P(\varepsilon)$ and $S(0)$ still intersect transversally at a point $A(\varepsilon) = \bar{A}$.

Remark 2.4.3. *The Implicit Function Theorem implies that the solution to the equation after perturbation is still 'generic', and Lemma 2.4.2 also asserts that the new manifolds intersect transversally, too.*

Chapter 3

The λ-μ-Structured Inverse Eigenvalue Problem

In this chapter we introduce a generalization of the λ-SIEP, which we call it the λ-μ-structured inverse eigenvalue problem. The problem asks about the existence of a $n \times n$ real symmetric matrix whose graph, spectrum, and the spectrum of one of its principal submatrices of order $n-1$ are prescribed. The results of this chapter are from [46].

3.1 The λ-μ-SIEP for trees

The first order Cauchy interlacing inequalities (1.2) assert that $\sigma(A(w))$ interlaces $\sigma(A)$, for any real symmetric matrix A and any fixed index w.

Problem 3. ***The λ-μ-SIEP for trees:*** *Two sets of real numbers $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$, $M = \{\mu_1, \mu_2, \dots, \mu_{n-1}\}$ and a tree T on n vertices $1, 2, \dots, n$ are given, such that M strictly interlaces Λ. For a fixed $w \in \{1, 2, \dots, n\}$ does there exist a real symmetric matrix A such that $\mathcal{G}(A) = T$, $\sigma(A) = \Lambda$, and $\sigma(A(w)) = M$?*

Below, we provide the sketch of a constructive proof for the solution to the λ-μ problem for trees. The proof is due to Duarte [3]. Then in Section 3.2 we show that this solution is 'generic', and in Section 3.3 we use the Jacobian method to show the existence of a solution for connected graphs.

Problem 4. ***The λ-μ-SIEP for connected graphs:*** *Two sets of real numbers $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$, $M = \{\mu_1, \mu_2, \ldots, \mu_{n-1}\}$ and a connected graph G on n vertices $1, 2, \ldots, n$ are given, such that M strictly interlaces Λ. For a fixed $w \in \{1, 2, \ldots, n\}$ does there exist a real symmetric matrix A such that $\mathcal{G}(A) = G$, $\sigma(A) = \Lambda$, and $\sigma(A(w)) = M$?*

The following theorem shows that the λ-μ-SIEP has finitely many solutions when G is a tree. The idea and the proof come from Duarte's 1989 paper [3].

Theorem 3.1.1. *Let T be a tree with vertices $1, 2, \ldots, n$ with $n \geq 2$, w be a fixed vertex and $\lambda_1, \ldots, \lambda_n, \mu_1, \ldots, \mu_{n-1}$ be real numbers satisfying*

$$\lambda_i < \mu_i < \lambda_{i+1}, \tag{3.1}$$

for all $i = 1, \ldots, n-1$. Then there exists an $A \in \mathcal{S}(T)$ with the Duarte-property with respect to w such that the λ's are the eigenvalues of A and the μ's are the eigenvalues of $A(w)$.

Proof. The proof is by induction on n. If T has two vertices, then the matrix

$$A = \begin{bmatrix} \mu_1 & \sqrt{(\lambda_2 - \mu_1)(\mu_1 - \lambda_1)} \\ \sqrt{(\lambda_2 - \mu_1)(\mu_1 - \lambda_1)} & \lambda_1 + \lambda_2 - \mu_1 \end{bmatrix}$$

has eigenvalues λ_1 and λ_2, $A(2)$ has eigenvalue μ_1, and A has the Duarte-property with respect to vertex 2. Interchanging the rows of A, and then interchanging the columns, we obtain a matrix with the Duarte-property with respect to vertex 1 and the desired spectral conditions.

Assume $n > 2$ and proceed by induction. Let $v_1, \ldots, v_k$ be the vertices adjacent to w in T, let $g_1(x), g_2(x), \ldots, g_k(x)$ be monic polynomials such that the degree of g_i is the number of vertices of $T_{v_i}(w)$ $(i = 1, 2, \ldots, k)$ and

$$g_1(x)g_2(x) \cdots g_k(x) = \prod_{j=1}^{n-1}(x - \mu_j).$$

As in [3] it can be shown that there exists a real number a_{ww}, positive real numbers a_{wv_j} $(i = 1, 2, \ldots, k)$ and real, monic polynomials $h_1, \ldots, h_k$ such that

$$\frac{\prod_{i=1}^{n}(x-\lambda_i)}{\prod_{j=1}^{n-1}(x-\mu_j)} = (x - a_{ww}) - \sum_{i=1}^{n-1} \frac{a_{wv_j}^2 h_j(x)}{g_j(x)}. \tag{3.2}$$

Also, as in [3], it is possible to show that the roots of h_j are real and strictly interlace those of g_j for each j.

By the induction hypothesis, there exist symmetric matrices $A_1, \ldots, A_k$ such that $\mathcal{G}(A_j) = T_{v_j}(w)$, A_j has the Duarte-property with respect to vertex v_j, A_j's characteristic polynomial is $g_j(x)$ and the characteristic polynomial of $A_{j'}(v_j)$ is $h_j(x)$, for $j = 1, \ldots, k$.

Let $A = [a_{ij}]$ be the $n \times n$ matrix such that $A_{v_j}(w) = A_j$, a_{ww}, and $a_{wv_j} = a_{v_jw}$ $(j = 1, 2, \ldots, k)$ are the real numbers defined in (3.2), and all other entries of A are zero. Then $A \in \mathcal{S}(T)$ and, as in Duarte [3], A and $A(w)$ have the desired eigenvalues. Since (3.1) holds and each A_j has the Duarte-property with respect to v_j, A has the Duarte-property with respect to w. $\square$

Remark 3.1.2. *Note that in the above proof we find exactly one solution for the λ-μ-SIEP for trees. All the other solutions come from the following choices:*

- *We can choose either a negative or a plus sign for a_{wv_j} and a_{v_jw}, hence realizing any symmetric sign pattern.*
- *We can choose the eigenvalues for each branch at any vertex.*

Remark 3.1.3. *The proof of Theorem 3.1.1 sheds light on the seemingly complicated definition of a matrix with Duarte property with respect to a vertex. It implies that a matrix A has the Duarte property with respect to vertex v if and only if $\sigma(A(v))$ strictly interlaces $\sigma(A)$.*

We now show that a matrix with the Duarte-property has a special property, somewhat akin to the strong Arnold property [52].

Lemma 3.1.4. *Let A have the Duarte-property with respect to the vertex w, $\mathcal{G}(A)$ be a tree T, and X be a symmetric matrix such that*

(a) $I \circ X = O$,

(b) $A \circ X = O$, *and*

(c) $[A, X](w) = O$.

Then $X = O$.

Proof. The proof is by induction on the number of the vertices. Without loss of generality we can take $w = 1$. For $n \leq 2$, (a) and (b) imply that $X = O$.

Assume $n \geq 3$ and proceed by induction. The matrices A and X have the form

$$A = \left[\begin{array}{c|cccc} a_{11} & b_1^T & b_2^T & \cdots & b_k^T \\ \hline b_1 & A_1 & O & \cdots & O \\ b_2 & O & A_2 & \cdots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_k & O & O & \cdots & A_k \end{array}\right], \qquad X = \left[\begin{array}{c|cccc} 0 & u_1^T & u_2^T & \cdots & u_k^T \\ \hline u_1 & X_{11} & X_{12} & \cdots & X_{1k} \\ u_2 & X_{21} & X_{22} & \cdots & X_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_k & X_{k1} & X_{k1} & \cdots & X_{kk} \end{array}\right],$$

so that each b_i has exactly one nonzero entry and without loss of generality we take this to be in its first position. Thus the A_i's correspond to the $T_v(w)$'s.

The $(2, 2)$-block of $[A, X]$ is

$$b_1 u_1^T + [A_1, X_{11}] - u_1 b_1^T = O.$$

Thus $[A_1, X_{11}] = u_1 b_1^T - b_1 u_1^T$. Since b_1 has just one nonzero entry, the nonzero entries of $u_1 b_1^T - b_1 u_1^T$ lie in its first row or first column. Thus $[A_1, X_{11}](1) = O$. Since A_1 has the Duarte property with respect to its first row, A_1 and X_{11} satisfy the induction hypothesis, and thus $X_{11} = O$ and $u_1 b_1^T - b_1 u_1^T = O$. Since the first row of $u_1 b_1^T - b_1 u_1^T$ is a nonzero multiple of u_1^T, we conclude that u_1 is the zero vector. An analogous argument shows that each of $X_{22}, X_{33}, \ldots, X_{kk}, u_2, u_3, \ldots, u_k$ is zero.

Now consider the $(i+1, j+1)$-block of $[A, X]$, where $i \neq j$. By (c), $A_i X_{ij} = X_{ij} A_j$. Since A has the Duarte-property with respect to vertex 1, A_i and A_j have no common eigenvalue. So, by part (a) of Lemma 1.2.4, $X_{ij} = O$. Thus $X = O$. □

Remark 3.1.5. *Lemma 3.1.4 guarantees that the matrix A constructed in Theorem 3.1.1 is 'generic'. That is, we can use the Jacobian method to extend the result to its superpatterns.*

3.2 A polynomial map and its Jacobian matrix

The following will be the setting throughout the remainder of this chapter. We fix T to be a tree with vertices $1, 2, \ldots, n$ and edges $e_k = \{i_k, j_k\}$, for $k = 1, \ldots, n-1$. Also fix G to be a supergraph of T with m additional edges. Let $x_1, x_2, \ldots, x_{2n-1}, y_1, y_2, \ldots, y_m$ be independent indeterminates, and set

$$x = (x_1, x_2, \ldots, x_{2n-1}), \text{ and } y = (y_1, y_2, \ldots, y_m).$$

Define $M(x, y)$ to be the matrix with $2x_i$ in the (i, i) position for $i = 1, 2 \ldots, n$, x_{n+k} in the (i_k, j_k) and (j_k, i_k) positions, for $k = 1, 2, \ldots, n-1$, y_k in the (i_k, j_k) and (j_k, i_k) positions, where $\{i_k, j_k\}$ is an edge of G not in T, for $k = 1, 2, \ldots, m$ and zeros elsewhere. Set $N(x, y) = M(x, y)(w)$; that is, $N(x, y)$ is the principal submatrix obtained from $M(x, y)$ by deleting its w-th row and column. We abbreviate $M(x, y)$ and $N(x, y)$ to M and N when convenient. Note that $2x_i$ is used for the (i, i)-position just to make the exposition a bit easier in the proof of the next lemma.

Example 3.2.1. *Consider the tree T in Figure 1.1, and let G be the complete graph on 5 vertices. The adjacency matrix of T is*

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and thus

$$M = \left[\begin{array}{c|cccc} x_1 & x_6 & x_7 & y_1 & x_8 \\ \hline x_6 & x_2 & y_2 & x_9 & y_3 \\ x_7 & y_2 & x_3 & y_4 & y_5 \\ y_1 & x_9 & y_4 & x_4 & y_6 \\ x_8 & y_3 & y_5 & y_6 & x_5 \end{array}\right], \quad \textit{and} \quad N = \begin{bmatrix} x_2 & y_2 & x_9 & y_3 \\ y_2 & x_3 & y_4 & y_5 \\ x_9 & y_4 & x_4 & y_6 \\ y_3 & y_5 & y_6 & x_5 \end{bmatrix}.$$

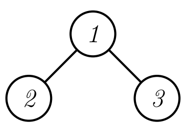
We now define two polynomial maps associated with M and N. Let $t^n + c_{n-1}(x,y)t^{n-1} + \cdots + c_1(x,y)t^1 + c_0(x,y)$ and $t^{n-1} + d_{n-1}(x,y)t^{n-2} + \cdots + d_1(x,y)t + d_0(x,y)$ be the characteristic polynomials of M and N, respectively. Also, let $g : \mathbb{R}^{2n-1} \times \mathbb{R}^m \to \mathbb{R}^{2n-1}$ be the polynomial map defined by

$$g(x,y) = (c_0(x,y), c_1(x,y), \ldots, c_{n-1}(x,y), d_0(x,y), d_1(x,y), \ldots, d_{n-2}(x,y)). \tag{3.3}$$

Let $f : \mathbb{R}^{2n-1} \times \mathbb{R}^m \to \mathbb{R}^{2n-1}$ be the polynomial map defined by

$$f(x,y) = \left(\frac{\operatorname{tr} M}{2}, \frac{\operatorname{tr} M^2}{4}, \ldots, \frac{\operatorname{tr} M^n}{2n}, \frac{\operatorname{tr} N}{2}, \frac{\operatorname{tr} N^2}{4}, \ldots, \frac{\operatorname{tr} N^{n-1}}{2(n-1)} \right). \tag{3.4}$$

Example 3.2.2. *Consider the matrices M and N for the tree T as shown:*



$$M = \begin{bmatrix} 2x_1 & x_4 & x_5 \\ x_4 & 2x_2 & y_1 \\ x_5 & y_1 & 2x_3 \end{bmatrix}, \qquad N = \begin{bmatrix} 2x_2 & y_1 \\ y_1 & 2x_3 \end{bmatrix}.$$

Then $f(x_1, x_2, x_3, x_4, x_5, y_1)$ equals

$$\begin{aligned} f(x_1, \ldots, x_5, y_1) = \Big(& x_1 + x_2 + x_3, x_1^2 + x_2^2 + x_3^2 + \frac{1}{2}(x_4^2 + x_5^2 + y_1^2), \\ & \frac{4}{3}(x_1^3 + x_2^3 + x_3^3) + x_1x_4^2 + x_1x_5^2 + x_2x_4^2 + x_2y_1^2 + x_3x_5^2 + x_3y_1^2 + x_4x_5y_1, \\ & x_2 + x_3, x_2^2 + x_3^2 + \frac{1}{2}y_1^2 \Big). \end{aligned}$$

By Newton's identities (1.2.2), there is an infinitely differentiable, invertible function $h : \mathbb{R}^{2n-1} \to \mathbb{R}^{2n-1}$ such that $g \circ h = f$. Thus, the Jacobian matrix of f at a point x is nonsingular if and only if the Jacobian matrix of g at $h(x)$ is nonsingular. In the next two results, we give a closed formula for the submatrix of the Jacobian matrix of the map f evaluated at A corresponding to the derivatives with respect to x_i's.

Lemma 3.2.3. *Let (i,j) be a nonzero position of M with corresponding variable x_t. Then*

(a) $\dfrac{\partial}{\partial x_t}\left(\operatorname{tr} M^k\right) = 2kM_{ij}^{k-1}$, *and*

(b) $\dfrac{\partial}{\partial x_t}\left(\operatorname{tr} N^k\right) = \begin{cases} 2kN_{ij}^{k-1} & \text{if neither } i \text{ nor } j \text{ is } n \\ 0 & \text{otherwise.} \end{cases}$

Proof. First, note that if $i \neq j$, then

$$\frac{\partial M}{\partial x_t} = E_{ij} + E_{ji},$$

and for $i = j$

$$\frac{\partial M}{\partial x_t} = 2E_{ii} = E_{ij} + E_{ji}.$$

Thus, in either case,

$$\begin{aligned}
\frac{\partial}{\partial x_t}\left(\operatorname{tr}(M^k)\right) &= \sum_{\ell=0}^{k-1} \operatorname{tr}\left(M^\ell \cdot \frac{\partial M}{\partial x_t} \cdot M^{k-\ell-1}\right) && \text{(by the chain rule)} \\
&= \sum_{\ell=0}^{k-1} \operatorname{tr}\left(M^{k-1} \cdot \frac{\partial M}{\partial x_t}\right) && \text{(since } \operatorname{tr}(AB) = \operatorname{tr}(BA) \text{ for any } A \text{ and } B\text{)} \\
&= k \operatorname{tr}\left(M^{k-1}(E_{ij} + E_{ji})\right) \\
&= k\left((M^{k-1})_{ij} + (M^{k-1})_{ji}\right) \\
&= 2k(M^{k-1})_{ij}. && \text{(since } M \text{ is symmetric)}
\end{aligned}$$

A similar argument works for N, provided we note that if i or j equals n then $\dfrac{\partial}{\partial x_t}N = 0$. □

Given a matrix $A = [a_{i,j}] \in \mathcal{S}(T)$ we denote by $\operatorname{Jac}(f)\big|_A$ the matrix obtained from $\operatorname{Jac}(f)$ by evaluating at $(x_1, \ldots, x_{2n-1}, y_1, \ldots, y_m)$ where x_k's and y_l's are equal to the corresponding entry of A for $k = 1, 2, \ldots, 2n-1$ and $l = 1, 2, \ldots, m$. Each column of $\operatorname{Jac}(f)\big|_A$ corresponds to the derivatives of f with respect to some x_i or some y_i. Let $\operatorname{Jac}(f)_x\big|_A$ denote the submatrix of $\operatorname{Jac}(f)\big|_A$ corresponding to those columns corresponding to the derivatives of f with respect to x_i's. Lemma 3.2.3 implies the following.

Corollary 3.2.4. *Let f be defined by 3.4, and let T be a tree, and $A \in \mathcal{S}(T)$. Then*

$$
\mathrm{Jac}_x(f)\Big|_A = \left[\begin{array}{ccc|ccc}
I_{11} & \cdots & I_{nn} & I_{i_1j_1} & \cdots & I_{i_{n-1}j_{n-1}} \\
A_{11} & \cdots & A_{nn} & A_{i_1j_1} & \cdots & A_{i_{n-1}j_{n-1}} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
A_{11}^{n-1} & \cdots & A_{nn}^{n-1} & A_{i_1j_1}^{n-1} & \cdots & A_{i_{n-1}j_{n-1}}^{n-1} \\
\hline
\widetilde{I}_{11} & \cdots & \widetilde{I}_{nn} & \widetilde{I}_{i_1j_1} & \cdots & \widetilde{I}_{i_{n-1}j_{n-1}} \\
\widetilde{B}_{11} & \cdots & \widetilde{B}_{nn} & \widetilde{B}_{i_1j_1} & \cdots & \widetilde{B}_{i_{n-1}j_{n-1}} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\widetilde{B}_{11}^{n-2} & \cdots & \widetilde{B}_{nn}^{n-2} & \widetilde{B}_{i_1j_1}^{n-2} & \cdots & \widetilde{B}_{i_{n-1}j_{n-1}}^{n-2}
\end{array}\right],
$$

where $\tilde{}$ *appends a zero row and a zero column to the matrix.*

The aim now is to show that the above matrix has full row rank, that is, it is nonsingular, whenever A has the Duarte-property with respect to n.

Theorem 3.2.5. *Let A, B and the function f be defined as above. If A has the Duarte-property with respect to vertex n, then* $\mathrm{Jac}(f)\Big|_A$ *has full row rank.*

Proof. Note that $\mathrm{Jac}(f)\Big|_A$ has full row rank if $\mathrm{Jac}_x(f)\Big|_A$ is nonsingular, and that happens if and only if the only vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{2n-1})^T$ such that $\alpha^T \mathrm{Jac}_x(f)\Big|_A = (0, \ldots, 0)$ is the zero-vector.

Let Jac_k denote the k-th row of $\mathrm{Jac}_x(f)\Big|_A$. So $\alpha^T \mathrm{Jac}_x(f)\Big|_A = \sum_{k=1}^{2n-1} \alpha_i \mathrm{Jac}_k$. Thus, for $\ell \leq n-1$, the ℓ-th entry in $\alpha^T \mathrm{Jac}_x(f)\Big|_A$ is the (i_ℓ, j_ℓ)-entry of $\sum_{k=0}^{n-1} \alpha_k A^k + \sum_{k=0}^{n-2} \alpha_{n+k} \widetilde{B}^k$, and for $\ell > n-1$ the ℓ-th entry in $\alpha^T \mathrm{Jac}_x(f)\Big|_A$ is the $(\ell-n+1, \ell-n+1)$-entry of $\sum_{k=0}^{n-1} \alpha_k A^k + \sum_{k=0}^{n-2} \alpha_{n+k} \widetilde{B}^k$.

Thus, we have shown that $\alpha^T \mathrm{Jac}_x(f)\Big|_A$ is the zero vector if and only if the matrix

$$X = \alpha_1 I + \alpha_2 A + \cdots + \alpha_n A^{n-1} + \alpha_{n+1} \widetilde{I} + \alpha_{n+2} \widetilde{B}^1 + \cdots + \alpha_{2n-1} \widetilde{B}^{n-2}$$

satisfies $X \circ A = O$ and $X \circ I = O$.

Let $p(x) = \sum_{i=1}^{n} \alpha_i x^{i-1}$ and $q(x) = \sum_{j=n+1}^{2n-1} \alpha_j x^{j-(n+1)}$. Then $X = p(A) + \widetilde{q(B)}$ and to show that $\mathrm{Jac}_x(A)$ is nonsingular it suffices to show that $p(x)$ and $q(x)$ are both zero polynomials.

$$
\begin{array}{r}
\alpha_1 \\ +\alpha_2 \\ \vdots \\ +\alpha_n \\ +\alpha_{n+1} \\ +\alpha_{n+2} \\ \vdots \\ +\alpha_{2n-1}
\end{array}
\left[\begin{array}{ccccc}
\cdots & I_{kk} & \cdots & I_{i_l j_l} & \cdots \\
\cdots & A_{kk} & \cdots & A_{i_l j_l} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\cdots & A^{n-1}_{kk} & \cdots & A^{n-1}_{i_l j_l} & \cdots \\
\hline
\cdots & \widetilde{I}_{kk} & \cdots & \widetilde{I}_{i_l j_l} & \cdots \\
\cdots & \widetilde{B}_{kk} & \cdots & \widetilde{B}_{i_l j_l} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\cdots & \widetilde{B}^{n-2}_{kk} & \cdots & \widetilde{B}^{n-2}_{i_l j_l} & \cdots
\end{array}\right]
$$

$$
= \quad \overline{[\cdots \quad 0 \quad \cdots \quad 0 \quad \cdots]}
$$

Figure 3.1: The product $\alpha^T \operatorname{Jac}_x(f)\Big|_A$.

Note that $[A, p(A)] = O$. Hence $[A, X] = [A, \widetilde{q(B)}]$. Also, note that since $A(n) = B$, $[A, \widetilde{q(B)}](n) = O$. Thus,$[A, X](n) = O$, and, by Lemma 3.1.4, we conclude that $X = O$. This implies that $p(A) = -\widetilde{q(B)}$. Let $Y := p(A) = -\widetilde{q(B)}$, then $AY = Ap(A)$. We claim that $Y = O$. Calculations yield:

$$
Ap(A) = -A(\widetilde{q(B)}) = -\left[\begin{array}{c|c} & * \\ B & * \\ & * \\ \hline * \cdots * & * \end{array}\right]\left[\begin{array}{c|c} & 0 \\ q(B) & \vdots \\ & 0 \\ \hline 0 \cdots 0 & 0 \end{array}\right] = \left[\begin{array}{c|c} & 0 \\ -Bq(B) & \vdots \\ & 0 \\ \hline * \cdots * & 0 \end{array}\right],
$$

and

$$
p(A)A = -\left[\begin{array}{c|c} & 0 \\ q(B) & \vdots \\ & 0 \\ \hline 0 \cdots 0 & 0 \end{array}\right]\left[\begin{array}{c|c} & * \\ B & \vdots \\ & * \\ \hline * \cdots * & * \end{array}\right] = \left[\begin{array}{c|c} & * \\ -q(B)B & \vdots \\ & * \\ \hline 0 \cdots 0 & 0 \end{array}\right].
$$

Since $Ap(A) = p(A)A$, the last row of $Ap(A)$ is zero and the last column of $Ap(A)$ is zero. Thus, $Ap(A) = -\widetilde{q(B)}\widetilde{B} = p(A)\widetilde{B}$. That is, $AY = Y\widetilde{B}$. Hence, by (a) of Lemma 1.2.4 either $Y = O$, or A and $\widetilde{B}$ have a common eigenvalue. If $Y = O$ we are done. Otherwise, since A and B have no common eigenvalue, A and $\widetilde{B}$ both have an eigenvalue 0 of multiplicity one.

Suppose column j of Y is nonzero, and let Y_j denote this column. Note the last entry of Y_j is 0. Since $AY = O$. By (b) of Lemma 1.2.4, Y_j is a generalized eigenvector of A

corresponding to 0. Since all of the eigenvalues of A have multiplicity 1, Y_j is an eigenvector of A, and it corresponds to the common eigenvalue 0. The form of A and the fact that the last entry of Y_j is 0 implies that the vector $Y_j(n)$ is a nonzero eigenvector of B corresponding to 0. This leads to the contradiction that A and B have a common eigenvalue. Thus $Y = O$.

Since $Y = O$, $p(A) = O$ and $q(B) = O$. Note that $p(x)$ is a polynomial of degree at most $n-1$. Since A has n distinct eigenvalues, its minimal polynomial has degree n. Thus $p(x)$ is the zero polynomial. Similarly $q(x)$ is the zero polynomial. So $\mathrm{Jac}_x(f)\big|_A$ is nonsingular, and consequently, $\mathrm{Jac}(f)\big|_A$ has full row rank. $\square$

3.3 The λ-μ-SIEP for connected graphs

We now use the Jacobian method to extend Theorem 3.1.1 to the supergraphs of trees, that is, connected graphs.

Theorem 3.3.1. *Let G be a connected graph with vertices $1, 2, \ldots, n$; i be a vertex of G, and $\lambda_1, \ldots, \lambda_n$, and $\mu_1, \ldots, \mu_{n-1}$ be real numbers satisfying (3.1). Then there is a symmetric matrix $A = \begin{bmatrix} a_{ij} \end{bmatrix}$ with graph G and eigenvalues $\lambda_1, \ldots, \lambda_n$ such that $A(i)$ has eigenvalues $\mu_1, \ldots, \mu_{n-1}$.*

Proof. Without loss of generality assume $i = n$. Let T be a spanning tree of G. Theorem 3.1.1 implies that there exists an $A \in \mathcal{S}(T)$ such that A has eigenvalues $\lambda_1, \ldots, \lambda_n$, $A(n)$ has eigenvalues $\mu_1, \ldots, \mu_{n-1}$, and A has the Duarte-property with respect to n. By Theorem 3.2.5, the Jacobian matrix of the function f defined by (3.4) evaluated at A has full row rank. Thus, the Jacobian matrix of the function g defined by (3.3) at A has full row rank.

Let $\boldsymbol{c}$ and $\boldsymbol{d}$ be the vectors of nonleading coefficients of the characteristic polynomials of A and $A(n)$, respectively. Let $\boldsymbol{a}$ be the assignment of the x_j's corresponding to A. We see that $g(\boldsymbol{a}, \boldsymbol{0}) = (\boldsymbol{c}, \boldsymbol{d})$. Since each of the last $n-1$ entries of $\boldsymbol{a}$ is nonzero, there is an open neighborhood U of $(\boldsymbol{a}, \boldsymbol{0})$ each of whose elements has no zeros in its first $n-1$ entries. By Theorem 2.1.1, there is an open neighborhood V of $\boldsymbol{a}$ and an open neighborhood W of $\boldsymbol{0}$ such that $V \times W \subseteq U$ and for each $\boldsymbol{y} \in W$ there is an $\boldsymbol{x} \in V$ such that $F(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{c}, \boldsymbol{d})$. Take $\boldsymbol{y}$ to be a vector in W with no zero entries. Then the $(\boldsymbol{x}, \boldsymbol{y})$ satisfying $F(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{c}, \boldsymbol{d})$

corresponds to a matrix $\bar{A} \in \mathcal{S}(G)$ such that the λ's are the eigenvalues of $\bar{A}$ and the μ's are the eigenvalues of $\bar{A}(n)$. $\square$

Remark 3.3.2. *Note that by proof of Theorem 3.3.1 for any real symmetric matrix A with graph G, if $\mathrm{Jac}(f)\big|_A$ has full row rank, then for every supergraph $\overline{G}$ of G, there is a matrix $\bar{A}$ whose graph is $\overline{G}$, and $\sigma(\bar{A}) = \sigma(A)$, and $\sigma(\overline{A}(w)) = \sigma(A(w))$. Furthermore, note that in the proof $\overline{A}$ can be taken to be arbitrarily close to A, entry-wise, hence keeping the eigenvalues of other submatrices of $\overline{A}$ arbitrarily close to those of A.*

Example 3.3.3. *Here we give a simple example to illustrate how this method works. Let $G = K_3$ and $i = 1$. We want to construct a 3×3 matrix A with prescribed eigenvalues, say -10, 0 and 2 such that the eigenvalues of $A(1)$ are prescribed and interlace those of A, say -1 and 1, and $\mathcal{G}(A) = K_3$. First, we choose an spanning tree of G and apply Duarte's method on it to realize the given spectral data.*

Figure 3.2: The graph C_3 and a spanning tree of it.

The adjacency matrix of T is

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Let

$$\hat{A} = \begin{bmatrix} a & d & e \\ d & b & 0 \\ e & 0 & c \end{bmatrix}.$$

Since $A(1)$ is going to be a diagonal matrix with eigenvalues $-1, 1$, we have $b = -1, c =$

1. *Also, we want the characteristic polynomial of A to satisfy*

$$C_A(\lambda) = (\lambda + 10)(\lambda)(\lambda - 2) = \lambda^3 + 8\lambda^2 - 20\lambda,$$

and

$$g(\lambda) = g_1(\lambda)g_2(\lambda) = (\lambda + 1)(\lambda - 1) = \lambda^2 - 1.$$

Then

$$\frac{C_A(\lambda)}{g(\lambda)} = \lambda - (-\mathbf{8}) - \left(\frac{\mathbf{27}}{\mathbf{2}}\frac{1}{\lambda+1} + \frac{\mathbf{11}}{\mathbf{2}}\frac{1}{\lambda-1}\right).$$

So, $a = \mathbf{-8}$, $d = \frac{\mathbf{27}}{\mathbf{2}}$, *and* $e = \frac{\mathbf{11}}{\mathbf{2}}$. *thus*

$$A = \begin{bmatrix} -8 & \sqrt{\frac{27}{2}} & \sqrt{\frac{11}{2}} \\ \sqrt{\frac{27}{2}} & -1 & 0 \\ \sqrt{\frac{11}{2}} & 0 & 1 \end{bmatrix}$$

realizes the given spectral data, and has the Duarte property. Let the matrices M and N and the function f be as described in Example 3.2.2. We calculate the Jacobian of f:

$$\mathrm{Jac}_x(f) = \left[\begin{array}{ccc|cc} 1 & 1 & 1 & 0 & 0 \\ 2x_1 & 2x_2 & 2x_3 & x_4 & x_5 \\ 4x_1^2 + x_4^2 + x_5^2 & 4x_2^2 + x_4^2 + y_1^2 & 4x_3^2 + x_5^2 + y_1^2 & 2x_1x_4 + 2x_2x_4 + x_5y_1 & 2x_1x_5 + 2x_3x_5 + x_4y_1 \\ \hline 0 & 1 & 1 & 0 & 0 \\ 0 & 2x_2 & 2x_3 & 0 & 0 \end{array}\right].$$

By direct calculation $\det(\mathrm{Jac}(f)) = 4\,x_2^2x_4x_5 - 8\,x_2x_3x_4x_5 - 2\,x_2x_4^2y_1 + 2\,x_2x_5^2y_1 + 4\,x_3^2x_4x_5 + 2\,x_3x_4^2y_1 - 2\,x_3x_5^2y_1$, *and*

$$\det(\mathrm{Jac}(f)\big|_A) = 24\sqrt{33} \neq 0.$$

So, the Implicit Function Theorem tells us that if we change the zero entries to some small nonzero number, there will be numbers close to the entries of A such that the new matrix constructed with these new numbers realizes the given spectral data. For example if $y = \sqrt{3}/2$, *then the matrix*

$$\bar{A} = \begin{bmatrix} -8 & \frac{9+\sqrt{11}}{2\sqrt{2}} & \frac{\sqrt{66}-3\sqrt{6}}{4} \\ \frac{9+\sqrt{11}}{2\sqrt{2}} & -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ \frac{\sqrt{66}-3\sqrt{6}}{4} & \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix}$$

has eigenvalues -10, 0 *and* 2 *and*

$$\bar{A}(1) = \begin{bmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix}$$

has eigenvalues -1 *and* 1.

Remark 3.3.4. *Note that Duarte's result (Theorem 3.1.1) solves the λ-SIEP for trees, since for given $\lambda_1 < \lambda_2 < \cdots < \lambda_n$ it is easy to find $\mu_1, \mu_2, \ldots, \mu_{n-1}$ such that (3.1) holds. Thus Theorem 3.3.1 immediately implies Theorem 2.3.4.*

3.4 A Geometric Interpretation

Similar to Section 2.4, in this section we introduce manifolds related to the λ-μ-SIEP, and study their tangent spaces and normal spaces at a point, and develop conditions for them to intersect transversally at that point.

Fix G, a connected graph on n vertices, $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ a set of n distinct real numbers, $M = \{\mu_1, \ldots, \mu_n\}$ a set of $n-1$ distinct real numbers, and v a fixed vertex of G. Furthermore, assume that M strictly interlaces Λ and fix a spanning tree T of G. Let A be the matrix given by Theorem 3.1.1 such that $\mathcal{G}(A) = T$, $\sigma(A) = \Lambda$, and $\sigma(A(v)) = M$.

Let

$$P_1 = \{B \in \mathrm{Sym}_n(\mathbb{R}) \,|\, \mathcal{G}(B) \text{is a subgraph of } T\},$$

$$P_2 = \{B \in \mathrm{Sym}_{n-1}(\mathbb{R}) \,|\, \mathcal{G}(B) \text{is a subgraph of } T(v)\},$$

and

$$S_1 = \{B \in \mathrm{Sym}_n(\mathbb{R}) \,|\, \sigma(B) = \Lambda\},$$

$$S_2 = \{B \in \mathrm{Sym}_{n-1}(\mathbb{R}) \,|\, \sigma(B) = M\}.$$

Also let

$$P = P_1 \times P_2,$$

and

$$S = S_1 \times S_2.$$

Then $(A, A(v)) \in P \cap S$. We want to perturb P to $P(\varepsilon)$ and find a point in the intersection of $P(\varepsilon)$ and S, using Lemma 2.4.2. So, we need to show that P and S intersect transversally at $(A, A(v))$. Using the following lemma [53], it is sufficient to show that both P_1 and S_1 intersect transversally at A, and P_2 and S_2 intersect transversally at $A(v)$.

Lemma 3.4.1. *For any two manifolds X and Y and points $p \in X$ and $q \in Y$,*

$$T_{(p,q)}(X \times Y) = T_p(X) \times T_q(Y).$$

Sketch of proof. Consider the canonical projections π_X and π_Y from $X \times Y$ to X and Y, respectively, and the map

$$f : T_{(p,q)}(X \times Y) \to T_p(X) \times T_q(Y)$$

$$v \mapsto \Big(d_{(p,q)}(\pi_X)(v), d_{(p,q)}(\pi_Y)(v)\Big).$$

It is easy to check that f is a linear map which is an isomorphism between $T_{(p,q)}(X \times Y)$ and $T_p(X) \times T_q(Y)$. □

Corollary 3.4.2. *Let M_1, M_2, N_1, and N_2 be manifolds such that M_1 and N_1 intersect transversally at a point p, and M_2 and N_2 intersect transversally at a point q. Then $M_1 \times M_2$ and $N_1 \times N_2$ intersect transversally at (p, q).*

In order to show that P_1 and S_1 intersect transversally at A we will show that their normal spaces at A intersect trivially. The case for P_2 and S_2 follows similarly. Let $N_a(M)$ denote the normal space to the manifold M at point a. Note that P_1 is a vector space. So, the tangent space to P_1 at any point is itself. Thus

$$N_A(P_1) = \{B \in \mathrm{Sym}_n(\mathbb{R}) \mid B \circ A = O, B \circ I = O\}.$$

Recall from Section 2.4 that

$$T_A(S_1) = \{KA - AK \mid K \text{ is skew-symmetric}\}.$$

Thus

$$N_A(S_1) = \{B \in \mathrm{Sym}_n(\mathbb{R}) \mid \operatorname{tr}(B(KA - AK)) = 0, \text{for all skew-symmetric matrices } K\}.$$

That is, for every B, an element of $N_A(S_1)$, we have

$$\begin{aligned}
\operatorname{tr}(B(KA - AK)) &= \operatorname{tr}(BKA - BAK) \\
&= \operatorname{tr}(ABK - BAK) \\
&= \operatorname{tr}((AB - BA)K) \\
&= 0,
\end{aligned}$$

for all $n \times n$ real skew-symmetric matrices K. That is, $AB - BA$ is perpendicular to the set of all $n \times n$ real skew-symmetric matrices. Hence

$$\begin{aligned}
N_A(S_1) &= \{B \in \operatorname{Sym}_n(\mathbb{R}) \mid BA - AB \text{ is perpendicular to all skew-symmetric matrices}\} \\
&= \{B \in \operatorname{Sym}_n(\mathbb{R}) \mid BA - AB \text{ is symmetric}\}.
\end{aligned}$$

But $BA - AB$ is skew-symmetric for all symmetric matrices A and B. Thus, $AB - BA = O$, if $B \in N_A(S_1)$. That is $N_A(S_1) = \{B \in \operatorname{Sym}_n(\mathbb{R}) \mid AB = BA\}$ is the centralizer of A in $\operatorname{Sym}_n(\mathbb{R})$. On the other hand, since λ_i's are distinct, the centralizer of A in $\operatorname{Sym}_n(\mathbb{R})$ is the set of all polynomials in A [54], that is, $N_A(S_1) = \{p(A) \in \operatorname{Sym}_n(\mathbb{R}) \mid p \in \mathbb{R}[x]\}$.

Recall that M strictly interlaces Λ. Thus, by Remark 3.1.3 A has the Duarte property with respect to vertex v. Thus, Lemma 3.1.4 implies that the only real symmetric matrix X such that $X \circ I = O$, $X \circ A = O$, and $XA - AX = O$ is the zero matrix. Hence, $N_A(S_1) \cap N_A(P_1) = \{O\}$. Therefore, S_1 and P_1 intersect transversally at A.

Note that since A has the Duarte property with respect to v, by definition $A(v)$ has the Duarte property with respect to each vertex $w \in \mathcal{N}(v)$. Thus, S_2 and P_2 intersect transversally at $A(v)$, similarly. Consequently, S and P intersect transversally at $(A, A(v))$, by Corollary 3.4.2.

By perturbing A to $\widehat{A}$ in the directions orthogonal to P we can realize any supergraph of T by $\widehat{A} \in P(\varepsilon)$, and then Lemma 2.4.2 guarantees that $P(\varepsilon)$ and $S(0)$ still intersect transversally at a point $A(\varepsilon) = \bar{A}$.

Chapter 4

The λ-τ-Structured Inverse Eigenvalue Problem

The λ-μ-SIEP (Problem 3) asks if, given a graph G of order n, real numbers $\lambda_1 \leq \cdots \leq \lambda_n$, real numbers $\mu_1 \leq \cdots \leq \mu_{n-1}$ and $i \in \{1, 2, \ldots, n\}$, does there exist a real symmetric matrix A whose graph is G such that A has eigenvalues $\lambda_1, \ldots, \lambda_n$, and $A(i)$ has eigenvalues $\mu_1, \ldots, \mu_{n-1}$? In 1989 Duarte solved the problem for any matrix whose graph is a tree, under the assumption that the μ's interlace the λ's, and the μ's and λ's are distinct [3], and in Chapter 3 we proved the existence of a solution for the λ-μ-SIEP when the graph is any connected graph. One way to generalize this inverse eigenvalue problem is to ask if there is an analogue for an $n \times n$ matrix and one of its $(n-2) \times (n-2)$ submatrices.

Problem 5. ***The* λ-τ*-SIEP:*** *Given a graph G of order n, two sets of real numbers $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ and $T = \{\tau_1, \ldots, \tau_{n-2}\}$, and distinct i and j in $\{1, 2, \ldots, n\}$, does there exist a real symmetric matrix A with $\}(A) = G$ such that $\sigma(A) = \Lambda$ and $\sigma A(\{i, j\}) = T$?*

Throughout this the chapter we assume that the λ's and the τ's are distinct and that no λ_i and τ_j are equal. Under this assumption we define the λ-τ *sequence* to be $X = x_1, x_2, \ldots, x_{2n-2}$, where $x_1 < x_2 < \cdots < x_{2n-2}$, and $\{x_1, x_2, \ldots, x_{2n-2}\} = \{\lambda_1, \lambda_2, \ldots, \lambda_n\} \cup \{\tau_1, \tau_2, \ldots, \tau_{n-2}\}$.

In Section 4.1 we introduce some necessary conditions for the λ-τ-SIEP to have a solution when the graph is a tree. In Section 4.2 we show that the λ-τ problem for adjacent vertices

i and j and G being a tree has a solution whenever the λ's and τ's are distinct, and certain necessary conditions are met. This is done by reducing the λ-τ problem to two λ-μ problems [46]. In section 4.3 we use the Jacobian method, also used in [46], to extend the result to connected graphs. In Section 4.4 we extend the results of the previous two sections to the case when the vertices i and j are not adjacent. Finally, in Section 4.5 we use the old and new results to answer a question regarding the eigenvalues of matrix A and $\hat{A}$, where $\hat{A}$ is obtained from A by perturbing one or two diagonal entries.

4.1 Properties of the λ-τ sequence

In this section we derive several properties of the λ-τ sequence of an $n \times n$ symmetric matrix $A = \left[a_{kl}\right]$ and a principal submatrix $A(\{i,j\})$, when the graph of G is a tree. Throughout this section $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ denote the eigenvalues of A, and $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_{n-2}$ denote the eigenvalues of $A(\{i,j\})$.

4.1.1 Restrictions on the λ-τ sequence

The Cauchy interlacing inequalities introduced in Lemma 1.2.1 describe some restrictions on the eigenvalues of A and $A(\{i,j\})$.

Proposition 4.1.1. *Let A be an $n \times n$ real symmetric matrix, and $A(\{i,j\})$ a principal submatrix of A of order $n-2$. Then the eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ of A and the eigenvalues $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_{n-2}$ of $A(\{i,j\})$ satisfy*

$$\lambda_t \leq \tau_t \leq \lambda_{t+2}, \tag{4.1}$$

for $t = 1, 2, \ldots, n-2$.

We refer to the inequalities in (4.1) as the *second order Cauchy interlacing inequalities.* We say $(A, A(\{i,j\}))$ is a *nondegenerate* pair if all inequalities in (4.1) are strict and no λ_k and τ_l are equal. The inequalities (4.1) imply some properties about the the λ-τ sequence of $(A, A(\{i,j\}))$. First, we have the following.

Lemma 4.1.2. *Let $X = x_1, x_2, \ldots, x_{2n-2}$ be the λ-τ sequence of $(A, A(\{i,j\}))$. Assume that X is nondegenerate. Then no three consecutive x_i's are eigenvalues of A, and no three consecutive x_i's are eigenvalues of $A(\{i,j\})$.*

Proof. Consider λ_k, λ_{k+1}, and λ_{k+2}. By (4.1), $\lambda_k \leq \tau_k \leq \lambda_{k+2}$. Thus λ_k, λ_{k+1}, and λ_{k+2} do not occur consecutively in the λ-τ sequence. The case of the eigenvalues of $A(\{i,j\})$ is similar. □

If τ_k and τ_{k+1} occur consecutively in the λ-τ sequence X, then we say that (τ_k, τ_{k+1}) is a *τ-pairing.* If λ_k and λ_{k+1} occur consecutively in the λ-τ sequence X, then we say that $(\lambda_k, \lambda_{k+1})$ is a *λ-pairing.* Note by (4.1) that if $(\tau_k, \tau_{k+1}) = (x_l, x_{l+1})$ is a τ-pairing, then $x_{l-1} = \lambda_{k+1}$ and $x_{l+2} = \lambda_{k+2}$. Also, if $(\lambda_k, \lambda_{k+1}) = (x_l, x_{l+1})$ is a λ-pairing, then $x_{l-1} = \tau_{k-1}$ and $x_{l+2} = \tau_k$.

Lemma 4.1.3. *Let X be the λ-τ sequence of $(A, A(\{i,j\}))$, and assume that X is nondegenerate. The first (that is, the one with smallest x_i's) and the last pairings of X are λ-pairings.*

Proof. Suppose $(\tau_k, \tau_{k+1}) = (x_l, x_{l+1})$ is a τ-pairing of X. Then $x_{l-1} = \lambda_{k+1}$, and thus $\{x_1, x_2, \ldots, x_{l-1}\} = \{\lambda_1, \lambda_2, \ldots, \lambda_{k+1}\} \cup \{\tau_1, \tau_2, \ldots, \tau_{k-1}\}$ contains two more λ's than τ's. Hence, there is a λ-pairing in $\{x_1, x_2, \ldots, x_{l-1}\}$, that is, one that precedes (τ_k, τ_{k+1}). Similarly, there is a λ-pairing in X that follows (τ_k, τ_{k+1}). □

Additionally, we have the following.

Lemma 4.1.4. *Let X be a nondegenerate λ-τ sequence. For any two τ-pairings in X there is a λ-pairing between them, and for any two λ-pairings in X there is a τ-pairing between them.*

Proof. Consider two consecutive τ-pairings in X, $(\tau_k, \tau_{k+1}) = (x_r, x_{r+1})$ and $(\tau_l, \tau_{l+1}) = (x_s, x_{s+1})$. Then

$$\{x_{r+2}, x_{r+3}, \ldots, x_{x_{s-1}}\} = \{\lambda_{k+2}, \lambda_{k+3}, \ldots, \lambda_{l+1}\} \cup \{\tau_{k+2}, \tau_{k+3}, \ldots, \tau_{l-1}\}.$$

Thus, $\{x_{r+2}, x_{r+3}, \ldots, x_{x_{s-1}}\}$ has two more λ's than τ's, and hence contains a λ-pairing. Similarly, there is a τ-pairing between any two λ-pairings. □

Lemmas 4.1.2 – 4.1.4 give restrictions on the choice of eigenvalues of A and $A(\{i, j\})$. A simple way to summarize the lemmas is that the τ-pairings of A and $A(\{i, j\})$ *interlace* the λ-pairings. Next, we use this nice property of the pairings to partition X into two sets of desired sizes such that each set includes (strictly) interlacing λ's and τ's. The two sets will later be used to reduce the λ-τ problem to two λ-μ problems.

Lemma 4.1.5. *Let X be a nondegenerate λ-τ sequence with exactly k τ-pairings, and let r and s be positive integers such that $r + s = n$, and $r, s \geq k + 1$. Then X can be partitioned into two sets such that the first set has r λ's and $r - 1$ τ's, and the second set s λ's and $s - 1$ τ's. Furthermore, in each set the τ's and the λ's satisfy the first order Cauchy interlacing inequalities (1.2).*

Proof. We give an algorithm for constructing such a partition (B, C). Since there are k τ-pairings, by Lemmas 4.1.3 and 4.1.4 there are $k + 1$ λ-pairings. First, assign one of the elements in each pairing to B and the other one to C, arbitrarily. This, by Lemmas 4.1.3 and 4.1.4, results in two sets each with $k + 1$ λ's that are interlaced by k τ's. Let $X' = \{x'_1, x'_2, \ldots, x'_{2(n-k-1)}\} = X \setminus (B \cup C)$. Thus, as long as we assign both of x'_{2l-1} and x'_{2l} to B or both to C, the τ's in B (respectively C) will interlace the λ's in B (respectively C). Hence by assigning $r - k - 1$ of these pairs to B and the remaining $s - k - 1$ to C, we obtain a partition with the desired properties.

Consider two consecutive pairings in X. The portion of X between the two pairing is of one of the following forms:

$$\boldsymbol{\lambda_i} < \boldsymbol{\lambda_{i+1}} < \tau_i < \lambda_{i+2} < \tau_{i+1} < \lambda_{i+3} < \cdots < \tau_{r-1} < \lambda_{r+1} < \boldsymbol{\tau_r} < \boldsymbol{\tau_{r+1}},$$

or

$$\boldsymbol{\tau_i} < \boldsymbol{\tau_{i+1}} < \lambda_{i+2} < \tau_{i+2} < \lambda_{i+3} < \tau_{i+3} < \cdots < \lambda_r < \tau_r < \boldsymbol{\lambda_{r+1}} < \boldsymbol{\lambda_{r+2}}.$$

So, we can always assign pairs of consecutive elements of the form $\tau_j < \lambda_k$ or $\lambda_j < \tau_k$ to either sets and still the τ's interlace the λ's. We can assign enough such pairs to each set in order to get the correct sizes. Thus the claim holds, and the two sets satisfy the desired conditions. $\square$

Example 4.1.6. *Let $k = 2$, and*

$$X : \lambda_1 < \tau_1 < \lambda_2 < \lambda_3 < \tau_2 < \lambda_4 < \tau_3 < \tau_4 < \lambda_5 < \lambda_6 < \tau_5 < \tau_6 < \lambda_7 < \lambda_8.$$

Note that there are two τ-pairings in X. We want to partition X into two sets, B, C, of λ's interlaced by τ's, where $|B| = 5 \geq 2 \cdot 2 + 1$, and $|C| = 9 \geq 2 \cdot 2 + 1$. One choice is to assign the first element in each pairing to B and the second element to C. Hence,

$$B : \quad \lambda_2 < \quad \tau_3 < \quad \lambda_5 < \quad \tau_5 < \quad \lambda_7,$$

and

$$C : \quad \lambda_3 < \quad \tau_4 < \quad \lambda_6 < \quad \tau_6 < \quad \lambda_8.$$

Finally

$$X' : \lambda_1 < \tau_1 < \quad \tau_2 < \lambda_4.$$

Since B already has 5 elements, we assign the remaining two pairs in X' to C so that it has 9 elements. That is,

$$B : \quad \lambda_2 < \quad \tau_3 < \quad \lambda_5 < \quad \tau_5 < \quad \lambda_7,$$

and

$$C : \lambda_1 < \tau_1 < \quad \lambda_3 < \tau_2 < \lambda_4 < \quad \tau_4 < \quad \lambda_6 < \quad \tau_6 < \quad \lambda_8.$$

Example 4.1.7. *A concrete example is provided here. Let λ's be -6, -5, -2, -1, 3, and 4, and let τ's be -4, -3, 1, and 2. Then using the above procedure we can get two sequences of λ's and τ's such that the τ's strictly interlace the λ's in each sequence. One such partition is:*

$$-6, \mathbf{-4}, -2, \mathbf{1}, 3,$$

and

$$-5, \mathbf{-3}, -1, \mathbf{2}, 4,$$

where bold numbers are the τ's.

4.1.2 Graph restrictions

In this section, we shall show that in addition to the restrictions on the λ-τ sequence given in Lemmas 4.1.2 – 4.1.4, there are restrictions related to the underlying graph. Throughout the remainder of this section we assume that $A = \begin{bmatrix} a_{ij} \end{bmatrix}$ is a real $n \times n$ symmetric matrix, the graph of A is a tree T, and r and s are adjacent vertices in T. Removing the edge $\{r, s\}$ from T results in a graph with two connected components. We let V_r be the set of vertices in the connected component that contains r, and V_s be the set of vertices of the other connected component. We let $\alpha_1, \alpha_2, \ldots, \alpha_{i_r}$ be the vertices in V_r adjacent to r, and $\beta_1, \beta_2, \ldots, \beta_{i_s}$ be the vertices in V_s adjacent to s.

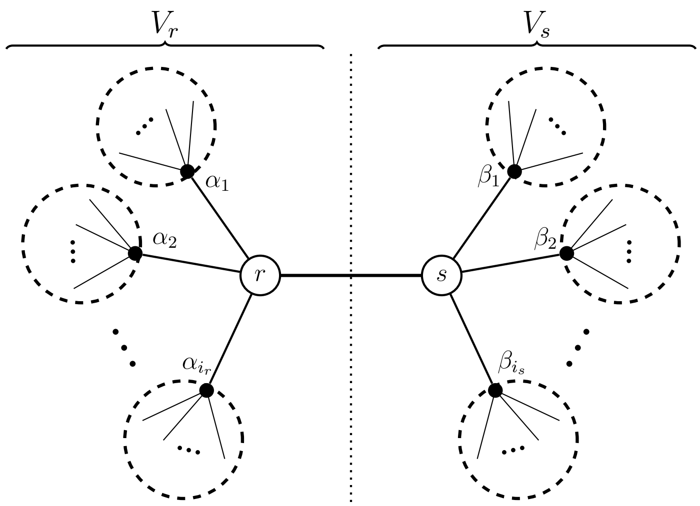


Figure 4.1: A tree T with adjacent vertices r and s, where other neighbors of r are $\alpha_1, \ldots, \alpha_{i_r}$, and other neighbors of s are $\beta_1, \ldots, \beta_{i_s}$.

The following lemma, which relates the characteristic polynomial of A to the characteristic polynomials of $A(\{r, s\})$, $A(r)$, and $A(s)$, plays a key role. For a detailed proof of the case A a zero-one matrix see [55, Proposition 5.1.1].

Lemma 4.1.8. *Let $A = \begin{bmatrix} a_{ij} \end{bmatrix}$ be a real symmetric $n \times n$ matrix whose graph is a tree T with vertices r and s adjacent. Let λ_i's be the eigenvalues of A and τ_i's be the eigenvalues of $A(\{r, s\})$. Then the characteristic polynomial of A is*

$$C_A(x) = -a_{rs}^2\, C_{A[V_r \setminus \{r\}]}(x) C_{A[V_s \setminus \{s\}]}(x) + C_{A[V_r]}(x) C_{A[V_s]}(x), \tag{4.2}$$

and
$$\frac{\prod_{i=1}^{n}(x-\lambda_i)}{\prod_{i=1}^{n-2}(x-\tau_i)} = -a_{rs}^2 + \frac{C_{A[V_r]}(x)}{C_{A[V_r\setminus\{r\}]}(x)}\frac{C_{A[V_s]}(x)}{C_{A[V_s\setminus\{s\}]}(x)}. \tag{4.3}$$

Proof. First, observe that since T is a tree, each nonzero term in $\det(xI-A)$ containing a_{rs} as a factor also contains a_{sr} as a factor. The first term of (4.2) represents the terms of the polynomial that contain a_{rs} as a factor, and the second term represents the terms that do not. Equation (4.3) is obtained by dividing both sides of (4.2) by $C_{A(\{r,s\})}$. □

When the λ-τ sequence of A is nondegenerate, the following shows that the cardinalities of V_r and V_s are upper bounds on the number of τ-pairings of this sequence.

Lemma 4.1.9. *Let A be an $n\times n$ real symmetric matrix with the property that its graph is a tree T, vertices r and s are adjacent in T, and the λ-τ sequence of $(A, A(\{r,s\}))$ is nondegenerate. If there are exactly k τ-pairings in the λ-τ sequence of $(A, A(\{r,s\}))$, then $|V_r|, |V_s| > k$.*

Proof. We claim that for each τ-pairing, one of the τ's is an eigenvalue of $A[V_r\setminus\{r\}]$ and the other one is an eigenvalue of $A[V_s\setminus\{s\}]$. Suppose to the contrary that both of the τ's in the pairing $\lambda_{i+1} < \tau_i < \tau_{i+1} < \lambda_{i+2}$ belong to $A[V_s\setminus\{s\}]$. Then τ_i and τ_{i+1} are also eigenvalues of $A(s)$, which by first order Cauchy interlacing inequalities should interlace the λ's. That is, there is a λ, an eigenvalue of A, such that $\tau_i < \lambda < \tau_{i+1}$. This contradicts our assumption that τ_i and τ_{i+1} form a pairing. Hence, each of the subgraphs $T[V_r\setminus\{r\}]$ and $T[V_s\setminus\{s\}]$ has one vertex for each τ-pairing, and we conclude that $|V_r|, |V_s| > k$. □

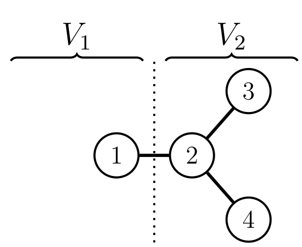


Figure 4.2: A star on four vertices

Example 4.1.10. *Let T be as in Figure 4.2, and A be a symmetric matrix with graph T. Then by Lemma 4.1.9 the λ-τ sequence of $(A, A(\{1,2\}))$ is not of the form $\lambda_1 < \lambda_2 < \tau_1 < \tau_2 < \lambda_3 < \lambda_4$, since $|V_1| = 1$, and there is a τ-pairing.*

The following lemma simply shows that for a rational function whose roots are all simple there is a sufficiently small vertical shift such that it does not change the number of roots between any two poles, and all the new roots are distinct from the old roots and the poles of the original function.

Lemma 4.1.11. *Let*

$$f(x) = \frac{\prod_{i=1}^{n}(x-\lambda_i)}{\prod_{i=1}^{n-2}(x-\tau_i)}, \tag{4.4}$$

where λ_i's and τ_i's satisfy the strict second order Cauchy interlacing inequalities and the λ-τ sequence is nondegenerate. Then for sufficiently small ε the function $f(x)+\varepsilon$ has exactly n distinct real roots, say $\mu_1, \mu_2, \ldots, \mu_n$, where $\mu_i \neq \tau_j$ for all i and j. Moreover, in the μ-τ sequence the μ's are exactly in the same position as λ's in the λ-τ sequence. That is, the τ_i's interlace the μ_i's in the same fashion that they interlace the λ_i's.

Proof. Since the λ-τ sequence is nondegenerate, $f(x)$ has exactly n roots λ_1,λ_2, …, λ_n, and all of them are simple roots. Furthermore, $f(x)$ is a continuous and differentiable function around its roots, so for each $i = 1, 2, \ldots, n$ there exist $\delta_i > 0$ such that $f'(x) \neq 0$ on the interval $[\lambda_i - \delta_i, \lambda_i - \delta_i]$. Let $\varepsilon = \frac{1}{2}\min\{|(f(\lambda_i \pm \delta_i)| : i = 1, \ldots, n\}$. □

Note that if $f(x)$ does not have any negative local extreme values, then ε can be chosen arbitrarily large. But since there is at least one λ-pairing, it can be shown that there is at least one negative local extreme value for the function f. Hence the choice of ε will be restricted to $(0, m)$, where m is the maximum of these negative local extreme values.

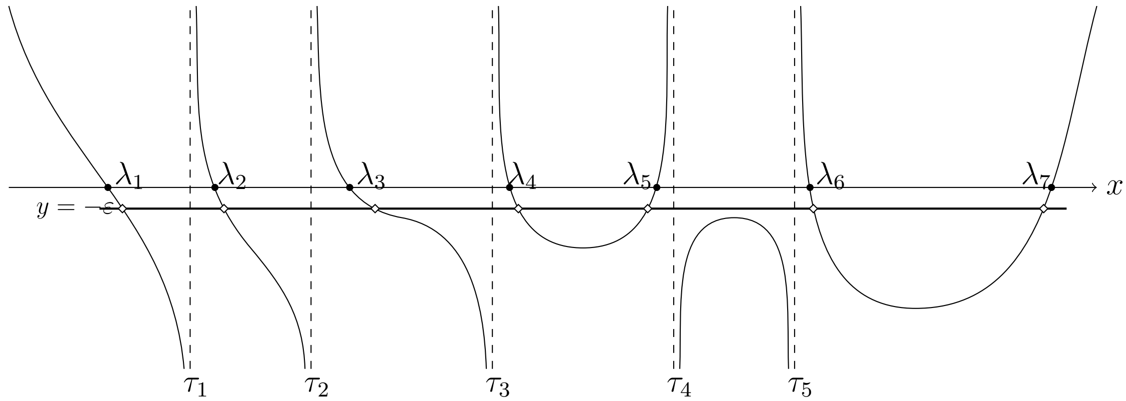


Figure 4.3: An example of the function f defined by (4.4), and a suitable choice of small enough ε.

4.2 The λ-τ structured inverse eigenvalue problem for trees

Recall that there is a $\varepsilon > 0$ such that τ_i's interlace the n real roots of $f(x) + \varepsilon$ in the same way that they interlace λ_i's. Below, the (r, s) entry of A is chosen such that $0 \leq a_{rs} \leq \sqrt{\varepsilon}$. Theorem 4.2.1 below shows that any such choice of a_{rs} is sufficient for solving the λ-τ problem for trees, provided the necessary conditions are satisfied. In other words, assuming nondegeneracy of the λ-τ sequence, the only constraints to solve a λ-τ structured inverse eigenvalue problem where the given graph is a tree and the vertices to be deleted are adjacent, are the second order Cauchy interlacing inequalities (Proposition 4.1.1) and the combinatorial restrictions (Lemma 4.1.9). Now we are ready to present and prove the main theorem for trees.

Theorem 4.2.1. *Let T be a tree with vertices $1, 2, \ldots, n$ such that its vertices r and s are adjacent, and $\lambda_1, \ldots, \lambda_n$, $\tau_1, \ldots, \tau_{n-2}$ be real numbers satisfying*

$$\lambda_i < \tau_i < \lambda_{i+2}, \tag{4.5}$$

and

$$\tau_i \neq \lambda_{i+1}, \tag{4.6}$$

for all $i = 1, \ldots, n-2$. Furthermore, assume that k τ-pairings occur, and $T[V_r \setminus \{r\}]$ and $T[V_s \setminus \{s\}]$ each have at least k vertices. Then there is a symmetric matrix $A = \left[a_{ij}\right]$ with graph T and eigenvalues $\lambda_1, \ldots, \lambda_n$ such that $A(\{r, s\})$ has eigenvalues $\tau_1, \ldots, \tau_{n-2}$.

Proof. Let T be a tree as in Figure 4.1 and $f(x)$ be defined by (4.4). By Lemma 4.1.11 there exists an $\varepsilon > 0$ such that $g(x) = f(x) + \varepsilon$ has n distinct real zeros. Let $a_{rs} = \sqrt{\varepsilon}$ and μ_1, μ_2, $\ldots$, μ_n be the roots of $g(x)$. For small enough $\varepsilon > 0$ the τ's interlace the μ's in the same way that τ's interlace λ's. Let X be the set of these μ's and τ's. Then X is nondegenerate with exactly k τ-pairings, $|V_r|$ and $|V_s|$ are positive integers such that $|V_r| + |V_s| = n$, and $|V_r|, |V_s| > k$. Thus, by Lemma 4.1.5 X can be partitioned into two sets X_1, X_2 such that X_1 has $|V_r|$ μ's and $|V_r| - 1$ τ's, and X_2 set has $|V_s|$ μ's and $|V_s| - 1$ τ's. Furthermore, in each set the τ's and the μ's satisfy first order Cauchy interlacing inequalities.

By Theorem 3.1.1 there are real symmetric matrices $A[V_r]$ and $A[V_s]$ such that the graph of $A[V_r]$ is $T[V_r]$ and the graph of $A[V_s]$ is $T[V_s]$. The set X_1 consists of the eigenvalues of $A[V_r]$ and $A[V_r \setminus \{r\}]$, and the set X_2 consists of the eigenvalues of $A[V_s]$ and $A[V_s \setminus \{s\}]$.

Now let $A = (A[V_r] \oplus A[V_s]) + a_{rs}(E_{rs} + E_{sr})$, where $E_{rs} + E_{sr}$ represents the matrix with 1's in the positions corresponding to the edge $\{r, s\}$ and zeros elsewhere. By Lemma 4.1.8 the eigenvalues of A are λ_i's and the eigenvalues of $A(\{r, s\})$ are τ_i's.

$$A = \left[\begin{array}{c|c} A[V_r] & O \\ & {\scriptstyle a_{rs}} \\ \hline {\scriptstyle a_{rs}} & \\ O & A[V_s] \end{array}\right].$$

□

We note that if $r = 1$ and $s = 2$, then by reordering the rows and the columns as

$(1, 2, \alpha_1, \ldots, \alpha_{i_1}, \beta_1, \ldots, \beta_{i_2})$, A has the form:

$$A = \left[\begin{array}{c|c|c|c} (A_\alpha)_{11} & a_{12} & A_\alpha[1,1) & 0 \quad \cdots \quad 0 \\ \hline a_{12} & (A_\beta)_{11} & 0 \quad \cdots \quad 0 & A_\beta[1,1) \\ \hline A_\alpha(1,1] & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} & A_\alpha(1) & O \\ \hline \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} & A_\beta(1,1] & O & A_\beta(1) \end{array}\right],$$

where $A_\alpha = A[V_1]$ and $A_\beta = A[V_2]$.

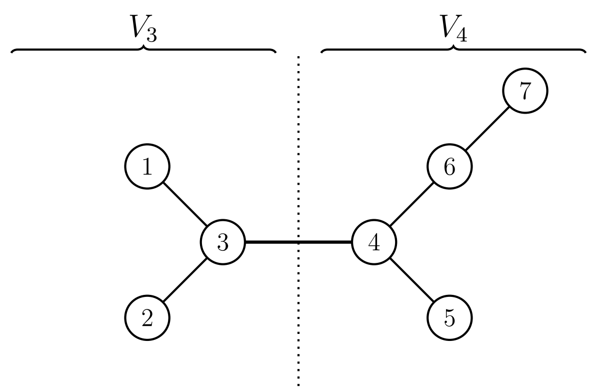


Figure 4.4: A tree T on 7 vertices with adjacent vertices 3 and 4.

Example 4.2.2. *Suppose we want to find a real symmetric matrix A whose graph is the tree T in Figure 4.4, such that its eigenvalues are $-6, -5, -2, -1, 3, 4,$ and 6, and the eigenvalues of $A(\{3, 4\})$ are $-4, -3, 1, 2,$ and 5. There are two τ-pairings, and they interlace the three λ-pairings. Note that $|V_3|, |V_4| > 2$. So, Theorem 4.2.1 guarantees the existence of such matrix A. To construct this matrix we choose $a_{3,4} = 1$, and find the roots μ_i of $f(x) + 1$ where f is defined by (4.4). Partition the μ-τ sequence into two sequences similar to the partitioning in Example 4.1.7. Following the algorithm given in the proof of Theorem 4.2.1 we find*

$$A \approx \left[\begin{array}{ccc|cccc} -4 & 0 & 2.292 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2.856 & 0 & 0 & 0 & 0 \\ 2.292 & 2.856 & -1.699 & \mathbf{1} & 0 & 0 & 0 \\ \hline 0 & 0 & \mathbf{1} & -0.3008 & 1.620 & 4.180 & 0 \\ 0 & 0 & 0 & 1.620 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4.180 & 0 & 0.2033 & 2.399 \\ 0 & 0 & 0 & 0 & 0 & 2.399 & -1.203 \end{array}\right].$$

Note that if we let $a_{3,4} = 0.1$ we get a different matrix

$$A \approx \left[\begin{array}{ccc|cccc} -4 & 0 & 2.366 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2.898 & 0 & 0 & 0 & 0 \\ 2.366 & 2.898 & -1.997 & \mathbf{0.1} & 0 & 0 & 0 \\ \hline 0 & 0 & \mathbf{0.1} & -0.002686 & 1.581 & 4.183 & 0 \\ 0 & 0 & 0 & 1.581 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4.183 & 0 & 0.2001 & 2.4 \\ 0 & 0 & 0 & 0 & 0 & 2.4 & -1.2 \end{array}\right].$$

It is easy to check that the graph of A is T and the eigenvalues of A and $A(\{3,4\})$ are as desired. Note that the approximations are because of machine error, and also approximations in finding the roots of the polynomials during the algorithm. Furthermore, note that we can choose all the off-diagonal entries corresponding to an edge of the graph to be positive.

4.3 The λ-τ-structured inverse eigenvalue problem for connected graphs where adjacent vertices are deleted

It is natural to ask if there is a result analogous to Theorem 4.2.1 for general connected graphs. First note that a connected graph which is not a tree has at least 3 vertices. So, for the rest of this section we can safely assume that $n \geq 3$. Here we use a similar approach to the one in Chapter 3 using the Implicit Function Theorem, the Duarte property, and a property similar to the Strong-Arnold hypothesis to give an affirmative answer to this question. Let A be a matrix whose graph is a tree T on n vertices $1, 2, \ldots, n$, with v and w adjacent.

Note that by construction, the matrices $A[V_r]$ and $A[V_s]$ in the Theorem 4.2.1 can be taken to have the Duarte property with respect to vertices r and s, respectively. Let $A = A[V_r] \oplus A[V_s]$, $\boldsymbol{x} = (x_1, x_2, \ldots, x_{2n-2})$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_p)$, where x_i's and y_j's are real variables, and $p = \frac{n^2-3n+4}{2}$.

Let $M(\boldsymbol{x}, \boldsymbol{y})$ be a matrix obtained from A by replacing diagonal entries by $2x_i$, $1 \leq i \leq n$, nonzero off-diagonal entries by x_{n+i}, $1 \leq i \leq n-2$, and zero off-diagonal entries by y_j, $1 \leq j \leq p$. Note that the entry corresponding to the edge $\{1, 2\}$ is now replaced by some y_j. Also define $N(\boldsymbol{x}, \boldsymbol{y}) := (M(\boldsymbol{x}, \boldsymbol{y}))(\{r, s\})$. We abbreviate $M(\boldsymbol{x}, \boldsymbol{y})$ and $N(\boldsymbol{x}, \boldsymbol{y})$ by M and N, respectively. Let $\boldsymbol{b} = (b_1, \ldots, b_p)$. For a function $f(\boldsymbol{x}, \boldsymbol{y})$ and a matrix $A = M(a_1, \ldots, a_{2n-1}, b_1, \ldots, b_p)$ we denote $f(a_1, \ldots, a_{2n-1}, b_1, \ldots, b_p)$ by $f(A, \boldsymbol{b})$. Similarly, $\mathrm{Jac}(f)\Big|_{(A,\boldsymbol{b})}$ denotes the Jacobian matrix of f where it is evaluated at $(\boldsymbol{x}, \boldsymbol{y}) = (a_1, \ldots, a_{2n-1}, b_1, \ldots, b_p)$.

Define $g : \mathbb{R}^{2n-2} \times \mathbb{R}^p \to \mathbb{R}^{2n-2}$ by

$$g(\boldsymbol{x}, \boldsymbol{y}) = (c_0, c_1, \ldots, c_{n-1}, d_0, d_1, \ldots, d_{n-3}),$$

where the c_i's and d_i's are the nonleading coefficients of the characteristic polynomials of M and N, respectively. We want to show that if $g(A, \mathbf{0}) = (\boldsymbol{c}, \boldsymbol{d}) \in \mathbb{R}^{2n-2}$ for some "generic" $(A, \mathbf{0}) \in \mathbb{R}^{2n-2} \times \mathbb{R}^p$, then for any sufficiently small perturbations $\boldsymbol{\varepsilon} \in \mathbb{R}^p$, there is an adjustment of $A \in \mathbb{R}^{2n-2}$, namely $\widehat{A}$, such that $g(\widehat{A}, \boldsymbol{\varepsilon}) = (\boldsymbol{c}, \boldsymbol{d})$. In other words, if the coefficients of the characteristic polynomials of A and $A(\{r, s\})$ are given by $\boldsymbol{c}$ and $\boldsymbol{d}$, respectively, then any superpattern of A has a realization with the same characteristic polynomial.

It is hard to work with partial derivatives of g. Using Newton's identities 1.2.2, we introduce a function f such that there exists a differentiable, invertible function h with $f \circ h = g$. Thus, similar to g, if $f(h(A, \mathbf{0})) = (\boldsymbol{a}, \boldsymbol{b})$, then there is a matrix $\hat{A}$ such that $f(h(\widehat{A}, \boldsymbol{\varepsilon})) = (\boldsymbol{a}, \boldsymbol{b})$.

Define the function $f : \mathbb{R}^{2n-2} \times \mathbb{R}^p \to \mathbb{R}^{2n-2}$ by

$$f(\boldsymbol{x}, \boldsymbol{y}) = \left(\frac{\operatorname{tr} M}{2}, \frac{\operatorname{tr} M^2}{4}, \ldots, \frac{\operatorname{tr} M^n}{2n}, \frac{\operatorname{tr} N}{2}, \frac{\operatorname{tr} N^2}{4}, \ldots, \frac{\operatorname{tr} N^{n-2}}{2(n-2)} \right). \tag{4.7}$$

Let $\mathrm{Jac}_x(f)$ be the matrix obtained from the Jacobian of f by deleting the columns corresponding to derivatives of f with respect to y_i's. We will show that $\mathrm{Jac}_x(f)$ evaluated at $(A, \mathbf{0})$ is nonsingular. The same calculations as in Lemma 3.2.3 yield the following:

Lemma 4.3.1. *Let M and N be as above and (i, j) be a nonzero position of M with corresponding variable x_t. Then*

(a) $\dfrac{\partial}{\partial x_t}\left(\operatorname{tr} M^k\right) = 2kM_{ij}^{k-1}$, *and*

(b) $\dfrac{\partial}{\partial x_t}\left(\operatorname{tr} N^k\right) = \begin{cases} 2kN_{ij}^{k-1} & \text{if neither } i \text{ nor } j \text{ is } 1 \text{ or } 2 \\ 0 & \text{otherwise.} \end{cases}$

Notation:

- For simplicity from now on let $r = 1$ and $s = 2$. Furthermore, assume that 1 is the first vertex of V_1, and 2 is the first vertex of V_2. Then for example, $A[V_2 \setminus \{2\}] = A[V_2](1)$
- $\widetilde{C}$ is a matrix obtained from a matrix C by appending two zero rows on top of it, and then two zero columns to left of the new matrix. That is,

$$\widetilde{C} = \left[\begin{array}{cc|ccc} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & & & \\ \vdots & \vdots & & \text{C} & \\ 0 & 0 & & & \end{array}\right].$$

- A '$*$' as an entry of a matrix means a real number whose value is not known.

Using Lemma 4.3.1 we can see the following.

Corollary 4.3.2. *Let f and A be defined as above. Then*

$$\operatorname{Jac}_x(f)\Big|_{(A,\mathbf{0})} = \left[\begin{array}{ccc|ccc} I_{i_1j_1} & \cdots & I_{i_{n-1}j_{n-1}} & I_{11} & \cdots & I_{nn} \\ A_{i_1j_1} & \cdots & A_{i_{n-1}j_{n-1}} & A_{11} & \cdots & A_{nn} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ A^{n-1}_{i_1j_1} & \cdots & A^{n-1}_{i_{n-1}j_{n-1}} & A^{n-1}_{11} & \cdots & A^{n-1}_{nn} \\ \hline \widetilde{I}_{i_1j_1} & \cdots & \widetilde{I}_{i_{n-1}j_{n-1}} & \widetilde{I}_{11} & \cdots & \widetilde{I}_{nn} \\ \widetilde{B}_{i_1j_1} & \cdots & \widetilde{B}_{i_{n-1}j_{n-1}} & \widetilde{B}_{11} & \cdots & \widetilde{B}_{nn} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \widetilde{B}^{n-3}_{i_1j_1} & \cdots & \widetilde{B}^{n-3}_{i_{n-1}j_{n-1}} & \widetilde{B}^{n-3}_{11} & \cdots & \widetilde{B}^{n-3}_{nn} \end{array}\right].$$

Note that this is a $(2n-2)\times(2n-2)$ matrix and in order to be nonsingular it suffices to show that it has full row rank.

The following theorem shows that the matrix $A = A[V_r] \oplus A[V_s]$ constructed in the proof of Theorem 4.2.1 is a 'generic' matrix. That is, the Jacobian of the function f defined by (4.7) evaluated at A is nonsingular.

Theorem 4.3.3. *Let T be a tree on n vertices $\{1, 2, \ldots, n\}$, where $\{r, s\}$ is an edge of T. Let A be a real symmetric matrix whose graph is $T \setminus \{r, s\}$, the function f be defined by (4.7), and $B = A(\{r, s\})$. If $A[V_r]$ has the Duarte-property with respect to vertex r, and $A[V_s]$ has the Duarte-property with respect to vertex s, then $\operatorname{Jac}_x(f)\Big|_{(A,\mathbf{0})}$ is nonsingular.*

Proof. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{2n-2})$ and assume that $\boldsymbol{\alpha}^T \operatorname{Jac}_x(f)\Big|_{(A,\mathbf{0})} = O$, that is,

$$\sum_{i=1}^{2n-2} \alpha_i \operatorname{Jac}(f)_k = O, \tag{4.8}$$

where $\operatorname{Jac}(f)_k$ denotes the k^{th} row of $\operatorname{Jac}_x(f)\Big|_{(A,\mathbf{0})}$. We want to show that $\boldsymbol{\alpha} = \mathbf{0}$.

Let $X = \alpha_1 I + \alpha_2 A + \cdots + \alpha_n A^{n-1} + \alpha_{n+1}\widetilde{I} + \alpha_{n+2}\widetilde{B}^1 + \cdots + \alpha_{2n-2}\widetilde{B}^{n-3}$. Note that each column of $\operatorname{Jac}_x(f)\Big|_{(A,\mathbf{0})}$ is evaluated only at a diagonal or a nonzero off-diagonal position of A. Thus $\boldsymbol{\alpha}^T \operatorname{Jac}_x(f)\Big|_{(A,\mathbf{0})} = O$ if and only if X is zero on all diagonal entries and all off-diagonal entries where A is nonzero, that is, $X \circ A = O$ and $X \circ I = O$.

We first show that $X = O$. Let

$$p(x) = \sum_{i=1}^{n} \alpha_i x^{i-1}, \text{ and } q(x) = \sum_{j=n+1}^{2n-1} \alpha_j x^{j-(n+1)}.$$

Then $X = p(A) + \widetilde{q(B)}$, and $\operatorname{Jac}_x(f)\Big|_{(A,\mathbf{0})}$ has full row rank if $p(x)$ and $q(x)$ are both zero polynomials. Since $[A, p(A)] = O$, $[A, X] = [A, \widetilde{q(B)}]$. Also, since $A(\{r,s\}) = B$, $[A, \widetilde{q(B)}](\{r,s\}) = O$. Hence, $[A, X](\{r,s\}) = O$.

Reorder rows and columns of A so that

$$A = \left[\begin{array}{cc|cc} C & x & & \\ \cline{1-2} x^T & * & \multicolumn{2}{c}{O} \\ \hline & & * & y^T \\ \cline{3-4} \multicolumn{2}{c|}{O} & y & D \end{array}\right],$$

where x and y correspond to vertices r and s, respectively. Then

$$\widetilde{q(B)} = \left[\begin{array}{cc|cc} q(C) & \begin{matrix}0\\ \vdots \\ 0\end{matrix} & & \\ \cline{1-2} 0 \ \cdots \ 0 & 0 & \multicolumn{2}{c}{O} \\ \hline & & 0 & 0 \ \cdots \ 0 \\ \cline{3-4} \multicolumn{2}{c|}{O} & \begin{matrix}0\\ \vdots \\ 0\end{matrix} & q(D) \end{array}\right].$$

By direct calculations we have

$$[A, \widetilde{q(B)}] = \left[\begin{array}{c|c} \begin{array}{ccc|c} & & & * \\ & O & & \vdots \\ & & & * \\ \hline * & \cdots & * & 0 \end{array} & O \\ \hline O & \begin{array}{c|ccc} 0 & * & \cdots & * \\ \hline * & & & \\ \vdots & & O & \\ * & & & \end{array} \end{array}\right].$$

Recall that the $(1,1)$ block of the above 2×2 block matrix corresponds to the indices in V_r, and its $(2,2)$ block corresponds to the indices in V_s. It follows that $[A\,[V_r]\,, X\,[V_r]](r) = O$ and $[A\,[V_s]\,, X\,[V_s]](s) = O$. Recall that the graphs of $A[V_1]$ and $A[V_2]$ are trees and these matrices are chosen to have the Duarte property with respect to the vertices 1 and 2, respectively. Thus, by Lemma 3.1.4 $X[V_r] = O$ and $X[V_s] = O$. So far it is shown that X has the following form:

$$X = \left[\begin{array}{c|c} O & X_1 \\ \hline X_2 & O \end{array}\right],$$

where the blocks are of the same size as the corresponding blocks in A. But X is a polynomial in A and $\widehat{B}$, hence X_1 and X_2 are also zero, hence, $X = O$. Thus, $p(A) = -\widetilde{q(B)}$. Let $Y = p(A) = -\widetilde{q(B)}$. Note that,

$$AY = Ap(A) = -A\widetilde{q(B)} = \left[\begin{array}{c|c|c|c} \begin{array}{ccc} & Cq(C) & \end{array} & \begin{array}{c} 0 \\ \vdots \\ 0 \end{array} & \multicolumn{2}{c}{O} \\ \hline * \quad \cdots \quad * & 0 & & \\ \hline & & 0 & * \quad \cdots \quad * \\ \cline{3-4} O & & \begin{array}{c} 0 \\ \vdots \\ 0 \end{array} & Dq(D) \end{array}\right],$$

and

$$YA = p(A)A = -\widetilde{q(B)}A = \left[\begin{array}{c|c|c|c} \begin{array}{ccc} & q(C)C & \end{array} & \begin{array}{c} * \\ \vdots \\ * \end{array} & \multicolumn{2}{c}{O} \\ \hline 0 \quad \cdots \quad 0 & 0 & & \\ \hline & & 0 & 0 \quad \cdots \quad 0 \\ \cline{3-4} O & & \begin{array}{c} * \\ \vdots \\ * \end{array} & q(D)D \end{array}\right].$$

Since $Ap(A) = p(A)A$, the stars are all zero. That is, $AY = Y\widetilde{B}$. Hence, by part (a) of Lemma 1.2.4 either $Y = O$ or A and $\widetilde{B}$ have a common eigenvalue. If $Y = O$ we are done. Otherwise, since A and B have no common eigenvalue, A and $\widetilde{B}$ both have an eigenvalue 0 and the multiplicity of it in A is 1. Suppose Y_j is a nonzero column of Y. Then, by part (b) of Lemma 1.2.4, Y_j is a generalized eigenvector of A corresponding to 0. But A has distinct eigenvalues, hence Y_j is an eigenvector of A corresponding to 0. Note that since $Y = -\widetilde{q(B)}$, $Y_j = \left[\begin{array}{cccc|cccc} * & \cdots & * & 0 & 0 & * & \cdots & * \end{array}\right]^T$, where the blocks are the same size as C and D. The form of A and Y_j imply that the vector $Y_j(\{r, s\})$ is a nonzero eigenvector of B corresponding to 0. This leads to a contradiction that A and B have a common eigenvalue. Thus $Y = O$.

Since $Y = O$, $p(A) = O$ and $q(B) = O$. Note that $p(x)$ is a polynomial of degree at most $n - 1$. Since A has n distinct eigenvalues, its minimal polynomial has degree n.

Thus $p(x)$ is the zero polynomial. Similarly $q(x)$ is the zero polynomial. So $\mathrm{Jac}_x(f)\Big|_{(A,\mathbf{0})}$ is nonsingular. $\square$

Now we are ready to prove an analogue to Theorem 4.2.1 for connected graphs.

Theorem 4.3.4. *Let G be a connected graph on n vertices $1, 2, \ldots, n$ with vertices 1 and 2 adjacent in G. Furthermore, assume that G has a spanning tree T containing the edge $\{1, 2\}$, and a partition $V_1 \cup V_2$ of its vertices with $|V_1|, |V_2| > k$ such that V_i contains vertex i for $i = 1, 2$. Let $\lambda_1, \ldots, \lambda_n$, $\tau_1, \ldots, \tau_{n-2}$ be real numbers satisfying*

$$\lambda_i < \tau_i < \lambda_{i+2}, \tag{4.9}$$

$$\tau_i \neq \lambda_{i+1}, \tag{4.10}$$

for all $i = 1 \ldots, n-2$. If k τ-pairings occur in the given λ-τ sequence, then there is a real symmetric matrix $A = \begin{bmatrix} a_{ij} \end{bmatrix}$ with graph G and eigenvalues $\lambda_1, \ldots, \lambda_n$ such that eigenvalues of $A(\{1, 2\})$ are $\tau_1, \ldots, \tau_{n-2}$.

Proof. Consider the spanning tree T of G. Theorem 4.2.1 implies that there exists an $A = (A[V_1] \oplus A[V_2]) + a_{12}(E_{12} + E_{21}) \in S(T)$ such that A has eigenvalues $\lambda_1, \ldots, \lambda_n$, $A(\{1, 2\})$ has eigenvalues $\tau_1, \ldots, \tau_{n-2}$, and $A[V_1], A[V_2]$ have the Duarte-property with respect to 1 and 2, respectively. By Theorem 4.3.3, the Jacobian matrix of the function f evaluated at A is nonsingular.

Let $\boldsymbol{c}$ and $\boldsymbol{d}$ be the vectors of nonleading coefficients of the characteristic polynomials of A and $A(\{1, 2\})$, respectively.

Letting $\boldsymbol{a} = (a_1, \ldots, a_n, a_{n+1}, a_{2n-2})$ be the assignment of the x_j's corresponding to A we see that $g(\boldsymbol{a}, 0, 0, \ldots, 0) = (c, d)$. Since $a_{n+1}, \ldots, a_{2n-2}$ are nonzero, there is an open neighborhood U of $(\boldsymbol{a}, 0, \ldots, 0)$ each of whose elements has no zeros in the same $n-2$ entries. By the Implicit Function Theorem 2.1.1, there is an open neighborhood V of $\boldsymbol{a}$ and an open neighborhood W of $\mathbf{0}$ such that $V \times W \subseteq U$ and for each $\boldsymbol{y} \in W$ there is an $\boldsymbol{x} \in V$ such that $F(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{c}, \boldsymbol{d})$. Take $\boldsymbol{y}$ to be a vector in W with no zero entries corresponding to the edges of G. Then the $(\boldsymbol{x}, \boldsymbol{y})$ satisfying $F(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{c}, \boldsymbol{d})$ corresponds to a matrix $\widehat{A} \in S(G)$ such that the λ's are the eigenvalues of $\widehat{A}$ and the τ's are the eigenvalues of $\widehat{A}(\{1, 2\})$. $\square$

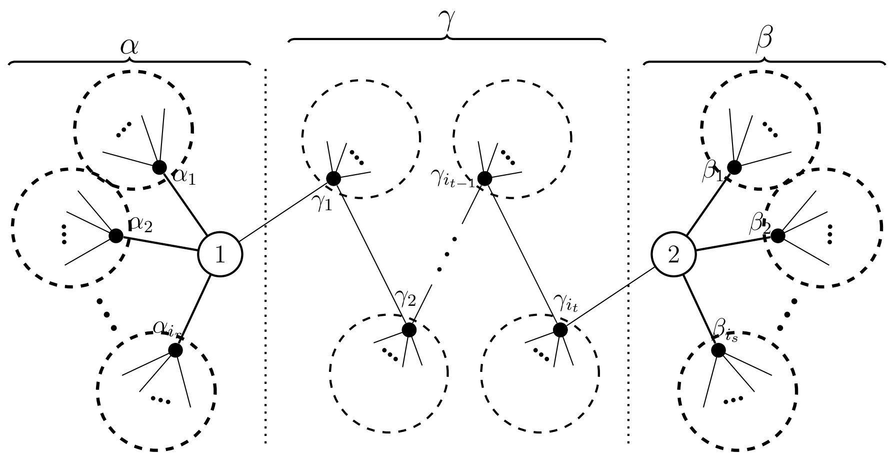


Figure 4.5: A tree T with non-adjacent vertices 1 and 2, where the neighbors of 1 are $\alpha_1, \ldots, \alpha_{i_r}, \gamma_1$, and other neighbors of s are $\beta_1, \ldots, \beta_{i_s}, \gamma_{i_t}$.

Remark 4.3.5. *Note that in the proof of Theorem 4.3.3 no conditions on the choice of a_{rs} are placed. For example, a_{rs} can be zero. Similarly, it could remain zero in the proof of Theorem 4.3.4. Hence, to prove Theorem 4.2.1 one could prove it for the forest obtained by deleting the edge $\{1, 2\}$, that is, by letting $a_{12} = 0$ in A. Then use Theorem 4.3.4 to extend it to the original tree, that is, a superpattern of the obtained forest.*

4.4 The case when the two removed vertices are not adjacent

In this section assume T is a tree on vertices $1, 2, \ldots, n$, where the vertices 1 and 2 are not adjacent. Let

$$\alpha = \left\{ v \;\middle|\; \begin{array}{l} \text{the path from } v \text{ to 1 does not contain} \\ \text{2, and the path from } v \text{ to 2 contains 1.} \end{array} \right\}$$

$$\beta = \left\{ v \;\middle|\; \begin{array}{l} \text{the path from } v \text{ to 2 does not contain} \\ \text{1, and the path from } v \text{ to 1 contains 2.} \end{array} \right\}$$

$$\gamma = \left\{ v \;\middle|\; \begin{array}{l} \text{the path from } v \text{ to 2 does not contain 1, and} \\ \text{the path from } v \text{ to 1 doesn not contain 2.} \end{array} \right\}$$

First note that in this case a variation of Lemma 4.1.9 holds.

Lemma 4.4.1. *Let A be an $n \times n$ real symmetric matrix whose graph is a tree T, as above. If there are exactly k τ-pairings in the eigenvalues of A and $A(\{1,2\})$, then $|\alpha|, |\beta| > k - |\gamma|$.*

Proof. The proof is similar to that of Lemma 4.1.9. That is, the τ's in a τ-pairing cannot both be eigenvalues of either $T[\alpha \setminus \{1\}]$ or $T[\beta \setminus \{2\}]$. Thus each τ is an eigenvalue of $T[\alpha \setminus \{1\}], T[\gamma]$, or $T[\beta \setminus \{2\}]$. In other words, for each τ-pairing $\lambda_{i+1}\tau_i < \tau_{i+1} < \lambda_{i+1}$; if neither τ_i nor τ_{i+1} belongs to $T[\beta \setminus \{2\}]$, then it belongs to $T[\gamma]$ or $T[\alpha \setminus \{1\}]$, hence $|\alpha| - 1 + |\gamma| \geq k$, and similarly $|\beta| - 1 + |\gamma| \geq k$. □

In order to solve a λ-τ problem for a tree where the deleted vertices are not adjacent we break T into two trees by deleting an edge in γ, and solve two λ-μ problems, similar to the ones of the Theorem 4.2.1. Then we show that this solution is generic in the same sense as in Theorem 4.3.3. Finally, we want to insert the deleted edge back to the tree and use the implicit function theorem to show that there is a solution. If one cannot divide γ into to parts γ_1, γ_2 such that $T[\alpha \cup \gamma_1]$ and $T[\beta \cup \gamma_2]$ are connected and each has at least $k+1$ vertices, then our method does not work. Hence, we assume that such a partition of γ exists:

Assumption 1. *There exist $\gamma_1, \gamma_2 \subseteq \gamma$ such that $\gamma_1 \cup \gamma_2 = \gamma$, $\gamma_1 \cap \gamma_2 = \emptyset$, and $T[\alpha \cup \gamma_1]$ and $T[\beta \cup \gamma_2]$ are connected and each has at least $k+1$ vertices.*

Theorem 4.4.2. *Let T be a tree on n vertices $1, 2, \ldots, n$ such that r and s are not adjacent, and $\lambda_1, \ldots, \lambda_n$, $\tau_1, \ldots, \tau_{n-2}$ real numbers satisfying*

$$\lambda_i < \tau_i < \lambda_{i+2}, \tag{4.11}$$

$$\tau_i \neq \lambda_{i+1}, \tag{4.12}$$

for all $i = 1 \ldots, n-2$. Furthermore, assume that there are k τ-pairings, and assumption 1 holds. Then there is a symmetric matrix $A = \begin{bmatrix} a_{ij} \end{bmatrix}$ with graph T and eigenvalues $\lambda_1, \ldots, \lambda_n$ such that $A(\{r,s\})$ has eigenvalues $\tau_1, \ldots, \tau_{n-2}$.

Proof. As T is a tree, there exists an edge $\{u, v\}$ on the path from r to s that divides γ into the two sets γ_r and γ_s. Partition the λ-τ sequence of $(A, A(\{r,s\}))$ into two sets X_r and X_s as in Lemma 4.1.5. By Theorem 4.2.1 there are matrices $A[\alpha \cup \gamma_r], A[\beta \cup \gamma_s]$ such that X_r consists of the eigenvalues of $A[\alpha \cup \gamma_r]$ and $A[\alpha \setminus \{r\} \cup \gamma_r]$ and X_s consists of the eigenvalues of $A[\beta \cup \gamma_s]$ and $A[\beta \setminus \{s\} \cup \gamma_s]$. Furthermore $\mathcal{G}(A[\alpha \cup \gamma_r]) = T[\alpha \cup \gamma_r]$, and $\mathcal{G}(A[\beta \cup \gamma_s]) = T[\beta \cup \gamma_s]$. Note that $A[\alpha \cup \gamma_r]$ can be taken to have the Duarte property with respect to vertex r, and $A[\beta \cup \gamma_s]$ can be taken to have the Duarte property with respect to vertex s.

Note that $a_{rs} = 0$. Let

$$A = \begin{array}{cc} \begin{array}{cc} \quad\quad\quad u & v \quad\quad\quad \end{array} & \\ \left[\begin{array}{c|c} A[\alpha \cup \gamma_1] & 0 \\ \hline 0 & A[\beta \cup \gamma_2] \end{array}\right] & \begin{array}{c} u \\ v \end{array} \end{array}$$

Let $G(A, 0) = (\boldsymbol{c}, \boldsymbol{d})$, where 0 comes from the (u, v) entry of A. By Theorem 4.3.3 the Jacobian of the function f defined in (4.7) evaluated at A is non-singular. Hence, for sufficiently small $\varepsilon > 0$ there is $\widehat{A}$ close to A, such that $G(\widehat{A}, \varepsilon) = (\boldsymbol{c}, \boldsymbol{d})$. That is, perturbing the (u, v) entry of A to be nonzero, the rest of the nonzero entries of A can be adjusted so that A and $A(\{r, s\})$ have the same characteristic polynomial as before, thus the same eigenvalues as before. $\square$

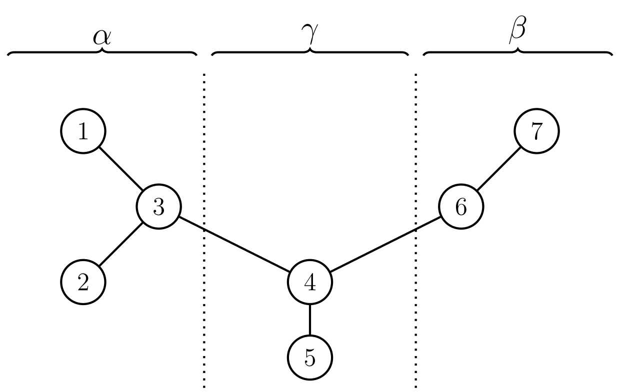


Figure 4.6: Tree T on 7 vertices where vertices 3 and 6 are not adjacent.

Example 4.4.3. *Suppose that we want to find a real symmetric matrix A whose graph is the tree T in Figure 4.6, such that its eigenvalues are $1, 2, 4, 6, 9, 10,$ and 12, and the eigenvalues of $A(\{3, 6\})$ are $3, 5, 7, 8,$ and 11. Note that there is one τ-pairing. In this method we need to delete an edge on the path from 3 to 6 and add the edge $\{3, 6\}$ and solve the problem for this tree, but we let $a_{3,6} = 0$. So we have two choices: case I: edge $\{3, 4\}$, and case II: edge $\{4, 6\}$. In either case $|V_3|, |V_6| > 1$. So, Theorem 4.4.2 guarantees the existence of such matrix A.*

Case I: $(r, s) = (3, 4)$.

Let T' be the tree obtained from T by deleting the edge $\{3, 4\}$ and inserting the edge $\{3, 6\}$.

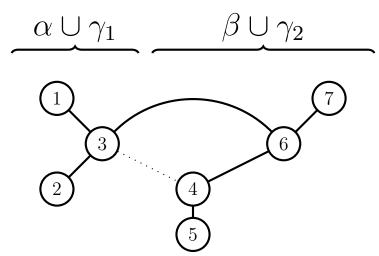


Figure 4.7: Tree T' where the edge $\{3, 4\}$ is removed and the edge $\{3, 6\}$ is added

Solve the problem for T', with 0 on $(3, 6)$ entry of A. Then use the Jacobian method to perturb $a_{3,4}$ to $\varepsilon = 0.1$, and adjust other entries to get the following matrix:

$$A \approx \begin{bmatrix} 3 & 0 & 1.732 & 0 & 0 & 0 & 0 \\ 0 & 7 & 3 & 0 & 0 & 0 & 0 \\ 1.732 & 3 & 4.003 & \mathbf{0.1} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{0.1} & 6.939 & 1.434 & 4.062 & 0 \\ 0 & 0 & 0 & 1.434 & 6.061 & 0 & 0 \\ 0 & 0 & 0 & 4.062 & 0 & 5.997 & 1.581 \\ 0 & 0 & 0 & 0 & 0 & 1.581 & 11 \end{bmatrix}.$$

Case II: $(r, s) = (4, 6)$.

Let T' be the tree obtained from T by deleting the edge $\{4, 6\}$ and inserting the edge $\{3, 6\}$.

Solve the problem for T', with 0 on the $(3, 6)$ entry of A. Then use the Jacobian method

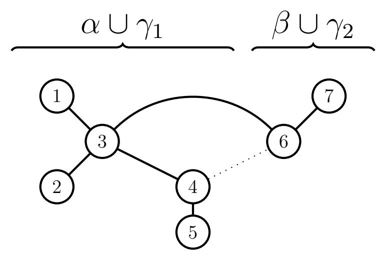


Figure 4.8: Tree T' where the edge $\{4,6\}$ is removed and the edge $\{3,6\}$ is added

to perturb $a_{4,6}$ *to* $\varepsilon = 0.1$*, and adjust other entries to get the following matrix:*

$$A \approx \begin{bmatrix} 7 & 0 & 2.372 & 0 & 0 & 0 & 0 \\ 0 & 11 & 1.909 & 0 & 0 & 0 & 0 \\ 2.372 & 1.909 & 6.004 & 3.119 & 0 & 0 & 0 \\ 0 & 0 & 3.119 & 3.960 & 0.999 & \mathbf{0.1} & 0 \\ 0 & 0 & 0 & 0.999 & 4.040 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{0.1} & 0 & 3.996 & 3.464 \\ 0 & 0 & 0 & 0 & 0 & 3.464 & 8 \end{bmatrix}.$$

It is easy to check that in both cases the graph of A *is* T *and the eigenvalues of* A *and* $A(\{3,4\})$ *are as desired. Note that the approximations are because of machine error, approximations in finding the roots of the polynomials during the algorithm, and also the error in the Newton's method we used to find the roots of the systems of multivariable polynomial equations. The number of iterations in the Newton's method to find above matrices is* 10.

Finally we state an analogue of Theorem 4.3.4 for the case of connected graphs.

Theorem 4.4.4. *Let* G *be a connected graph on* n *vertices* $1, 2, \ldots, n$ *where* r *and* s *are not adjacent, and let* $\lambda_1, \ldots, \lambda_n$*,* $\tau_1, \ldots, \tau_{n-2}$ *be real numbers satisfying*

$$\lambda_i < \tau_i < \lambda_{i+2}, \tag{4.13}$$

$$\tau_i \neq \lambda_{i+1}, \tag{4.14}$$

for all $i = 1 \ldots, n-2$*. Furthermore, assume that* k τ*-pairings occur. If* G *has a spanning* *tree T such that $|\alpha|, |\beta| \geq k - |\gamma|$, then there is a real symmetric matrix $A = \left[a_{ij}\right]$ with graph G and eigenvalues $\lambda_1, \ldots, \lambda_n$ such that eigenvalues of $A(\{r, s\})$ are $\tau_1, \ldots, \tau_{n-2}$.*

Proof. Let T be a spanning tree of G such that $|\alpha|, |\beta| \geq k - |\gamma|$. By Theorem 4.4.2, there is a matrix A with desired spectra whose graph is T. The rest of the proof exactly follows that of Theorem 4.3.4.

□

4.5 Diagonal perturbations

Deleting the i-th row and column of a matrix is closely related to perturbing the i-th diagonal entry of the matrix. In this section, the same ideas and techniques of the previous sections are used to study the spectra of a matrix and a diagonal perturbation of the matrix. We begin by studying the case that one diagonal entry is perturbed.

Lemma 4.5.1. *Let*

$$g(x) = (x - c_1)(x - c_2) \cdots (x - c_n),$$

$$h(x) = (x - d_1)(x - d_2) \cdots (x - d_n),$$

and

$$f(x) = g(x) - h(x),$$

where $d_1 < c_1 < d_2 < c_2 < \cdots < d_n < c_n$. Then f has exactly $n - 1$ real roots $e_1, \ldots, e_{n-1}$, and they satisfy the inequalities $d_1 < e_1 < d_2 < \cdots < e_{n-1} < d_n$.

Proof. First, note that the degree of f is $n - 1$, since the coefficient of x^{n-1} in f is the positive quantity $\sum_{i=1}^{n} c_i - d_i > 0$. Next, note that $g(d_i)$ and $g(d_{i+1})$ have opposite signs for all $i = 1, \ldots, n-1$. Consequently, $f(d_i)$ and $f(d_{i+1})$ have opposite signs for all $i = 1, \ldots, n-1$. Hence, f has a zero between d_i and d_{i+1} for all $i = 1, \ldots, n - 1$. That is, f has $n - 1$ roots e_i, such that $d_1 < e_1 < d_2 < \cdots < d_{n-1} < e_{n-1} < d_n$. □

Lemma 4.5.2. *Let A be a matrix, and let $\widehat{A} = A + aE_{ii}$, where E_{ii} is the matrix of the same size as A with its (i, i) entry equal to 1 and all other entries equal to zero. Then*

$$C_{\widehat{A}}(x) = C_A(x) + aC_{A(i)}(x).$$

Proof. This follows from the expansion of $\det(xI - \widehat{A})$ along the i-th row. $\square$

The above lemma implies that the eigenvalues of $A(i)$ are determined from those of $A+aE_{ii}$ and A. Thus, a solution to the λ-μ structured inverse eigenvalue problem defined by the eigenvalues of A and $A(i)$ is also a solution to the following related diagonal perturbation problem.

Theorem 4.5.3. *Let*

$$\lambda_1 < \mu_1 < \lambda_2 < \mu_2 < \cdots < \lambda_n < \mu_n$$

be $2n$ real numbers, and i an integer with $1 \leq i \leq n$. Given a tree T there is an $n \times n$ real symmetric matrix A whose graph is T, A has eigenvalues $\lambda_1, \ldots, \lambda_n$, and $A + aE_{ii}$ has eigenvalues $\mu_1, \ldots, \mu_n$, where $a = \sum_{j=1}^{n}(\mu_j - \lambda_j) > 0$.

Proof. Let

$$f(x) := \big((x-\mu_1)(x-\mu_2)\cdots(x-\mu_n)\big) - \big((x-\lambda_1)(x-\lambda_2)\cdots(x-\lambda_n)\big).$$

By Lemma 4.5.1 $f(x)$ has exactly $n-1$ real roots $\gamma_1 < \cdots < \gamma_{n-1}$ which strictly interlace λ_i's. That is,

$$\lambda_1 < \gamma_1 < \lambda_2 < \gamma_2 < \cdots < \gamma_{n-1} < \lambda_n.$$

So

$$f(x) = \left(\sum_{j=1}^{n}(\mu_j - \lambda_j)\right)\prod_{i=1}^{n-1}(x-\gamma_i) = a\prod_{i=1}^{n-1}(x-\gamma_i). \tag{4.15}$$

By Theorem 3.1.1 there is a real symmetric matrix A with the Duarte property with respect to vertex i whose graph is T, with eigenvalues $\lambda_1, \ldots, \lambda_n$ such that the eigenvalues of $A(i)$ are $\gamma_1, \ldots, \gamma_{n-1}$. Let $\widehat{A} = A + aE_{ii}$. By (4.15) we have

$$\begin{aligned} C_{\widehat{A}}(x) &= C_A(x) + a\,C_{A(i)}(x) \\ &= (x-\lambda_1)(x-\lambda_2)\cdots(x-\lambda_n) + a(x-\gamma_1)(x-\gamma_2)\cdots(x-\gamma_n) \\ &= (x-\mu_1)(x-\mu_2)\cdots(x-\mu_n), \end{aligned}$$

that is, the eigenvalues of $A + aE_{ii}$ are $\mu_1, \ldots, \mu_n$. $\square$

It is natural to ask if the above matrix A is 'generic' since the matrix obtained in Theorem 3.3.1 is. Below, we answer this question in the affirmative. We begin with the following technical lemma.

Lemma 4.5.4. *Let A be a real symmetric matrix whose graph is a tree T on vertices $1, 2, \ldots, n$, and let a be a real positive number. Assume that A has the Duarte property with respect to vertex 1. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_{2n-1}, y)$ and let $M = M(\boldsymbol{x})$ be defined as in Section 4.3, except for the $(1,1)$-entry which is x_1. Also let $\widehat{M} = \widehat{M}(\boldsymbol{x}) = M(\boldsymbol{x}) + yE_{11}$. Define the function $f : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ by*

$$f(\boldsymbol{x}) = (c_0, c_1, \ldots, c_{n-1}, d_0, d_1, \ldots, d_{n-1}),$$

where c_i and d_i are the nonleading coefficients of the characteristic polynomials of M and $\widehat{M}$, respectively. Then the Jacobian of f evaluated at $(A, \mathbf{0})$ is nonsingular.

Proof. Let $g : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ be defined by

$$g(\boldsymbol{x}) = (c_0, c_1, \ldots, c_{n-1}, e_0, e_1, \ldots, e_{n-2}),$$

where the c_i are the nonleading coefficients of the characteristic polynomials of M, and the e_i are the nonleading coefficients of the characteristic polynomial of $N = N(\boldsymbol{x}) = M(\boldsymbol{x})\,(1)$. As shown by Theorems 3.2.5 and 3.3.1, the Jacobian of g evaluated at A is nonsingular. Let

$$\operatorname{Jac}(g)\Big|_A = \left[\begin{array}{c} P_{n\times(2n-1)} \\ \hline Q_{(n-1)\times(2n-1)} \end{array}\right],$$

where the rows of P denote the derivatives of the c_i's evaluated at A, and the rows of Q denote the derivatives of the e_i's evaluated at A.

Observe that $C_{\widehat{M}}(x) = C_M(x) - yC_{M(1)}(x)$. Thus

$$\operatorname{Jac}(f)\Big|_{(A,a)} = \left[\begin{array}{c|c} & 0 \\ P & \vdots \\ & 0 \\ \hline & e_0 \\ P(\,;n) - aQ & \vdots \\ & e_{n-2} \\ \hline P_n & 1 \end{array}\right], \tag{4.16}$$

where P_n is the last row of P, and the e_i's are evaluated at (A, a). In the matrix in (4.16), subtract each row of P from the corresponding rows in the second and third block rows, and then scale the rows of the second block row by $\frac{1}{a}$. Now, the last row is $\left[\begin{array}{ccc|c} 0 & \cdots & 0 & 1 \end{array}\right]$. Subtract appropriate multiples of the last row from each row in the second block row to make all the entries of the last column of that block zero. The resulting matrix

$$\left[\begin{array}{c|c} & 0 \\ P & \vdots \\ & 0 \\ \hline & 0 \\ Q & \vdots \\ & 0 \\ \hline 0 \ \cdots \ 0 & 1 \end{array}\right],$$

which is row equivalent to $\operatorname{Jac}(f)\Big|_{(A,a)}$, is nonsingular, since $\left[\begin{array}{c} P \\ \hline Q \end{array}\right]$ is. $\square$

Theorem 4.5.5. *Let*

$$\lambda_1 < \mu_1 < \lambda_2 < \mu_2 < \cdots < \lambda_n < \mu_n$$

be $2n$ real numbers, and i an integer with $1 \le i \le n$. Given a connected graph G there is an $n \times n$ real symmetric matrix A such that the graph of A is G, A has eigenvalues $\lambda_1, \ldots, \lambda_n$, and $A + aE_{ii}$ has eigenvalues $\mu_1, \ldots, \mu_n$, where $a = \sum_{j=1}^{n}(\mu_j - \lambda_j) > 0$.

Sketch of the proof. Since G is a connected graph, it has a spanning tree T. By Theorem 4.5.3 there is an $n \times n$ real symmetric matrix A with the Duarte property with respect to vertex i such that the graph of A is T, A has eigenvalues $\lambda_1, \ldots, \lambda_n$, and $A + aE_{ii}$ has eigenvalues $\mu_1, \ldots, \mu_n$. By Lemma 4.5.4 the Jacobian of f evaluated at $(A, \mathbf{0})$ is nonsingular. Hence, by Theorem 2.1.1, for a sufficiently small perturbation $\boldsymbol{\varepsilon}$ of the zero entries of A corresponding to edges in $G \setminus T$, there are adjustments of the diagonal and nonzero off-diagonal entries of A to yield $\widehat{A}$ such that $f(\widehat{A}, \boldsymbol{\varepsilon}) = f(A, \mathbf{0})$. That is, if none of the entries of $\boldsymbol{\varepsilon}$ are zero and they are sufficiently small, then graph of $\widehat{A}$ is G, A has eigenvalues $\lambda_1, \ldots, \lambda_n$, and $A + aE_{ii}$ has eigenvalues $\mu_1, \ldots, \mu_n$. $\square$

Now we are ready to study perturbations involving two diagonal entries. Note that one cannot simply perturb one diagonal entry and then perturb another diagonal entry using the above method twice, since the matrix A given by Theorem 4.5.3 varies for each perturbation.

Theorem 4.5.6. *Let $\lambda_1, \ldots, \lambda_n$ and $\tau_1, \ldots, \tau_n$ be real numbers such that*

$$\lambda_i < \tau_i < \lambda_{i+2}, \tag{4.17}$$

$$\tau_i \neq \lambda_{i+1}, \tag{4.18}$$

for all $i = 1 \ldots, n-2$, and

$$\lambda_{n-1} < \tau_{n-1}, \lambda_n < \tau_n. \tag{4.19}$$

Assume a graph G satisfies the conditions of Theorems 4.3.4 or 4.4.4. Then there is a real symmetric matrix A and real numbers a_1 and a_2 such that the graph of A is G, the eigenvalues of A are the λ_i's, and the eigenvalues of $A + a_1E_{11} + a_2E_{22}$ are the τ_i's.

Proof. Let T be a spanning tree of G, and T' be the forest obtained from T by deleting the edge $\{u, v\}$ which satisfies the condition in the proof of Theorem 4.4.2. Note that in the case that 1 and 2 are adjacent in T, then $\{u, v\} = \{1, 2\}$. Call the two obtained connected components T_1 and T_2, where T_i contains vertex i. By Lemma 4.1.5 the set of all λ's and the smallest $n-2$ τ's can be partitioned into two sets of sizes at least $2|T_1|-1$ and $2|T_2|-1$, such that in each set the τ's interlace the λ's. There are two τ's left, which are the largest τ's. Assign each of them to one of the sets. By Theorem 4.5.3 each of these sets can be realized as eigenvalues of a matrix A_i and $A_i + a_i E_{11}$ with graph T_i where A_i has the Duarte property with respect to vertex i, for $i = 1, 2$. Let $A = A_1 \oplus A_2$ and $\hat{A} = A + a_1 E_{11} + a_2 E_{22}$. Let M be the matrix obtained from replacing each diagonal entry of A by $2x_j$, $1 \leq j \leq n$, and by replacing each nonzero off-diagonal entry by x_{n+j}, $1 \leq j \leq n-2$. Let $N := M + yE_{11} + zE_{22}$.

Define $f : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ with

$$f(x_1, x_2, \ldots, x_{2n-2}, y, z) = (c_0, c_1, \ldots, c_{n-1}, d_0, d_1, \ldots, d_{n-1}),$$

where c_i and d_i are the nonleading coefficients of M and N, respectively. Define $g : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ with

$$g(x_1, x_2, \ldots, x_{2n-2}, y, z) = \left(\frac{\operatorname{tr} M}{2}, \frac{\operatorname{tr} M^2}{4}, \ldots, \frac{\operatorname{tr} M^n}{2n}, \frac{\operatorname{tr} N}{2}, \frac{\operatorname{tr} N^2}{4}, \ldots, \frac{\operatorname{tr} N^n}{2n} \right).$$

Newton's identities imply that $\operatorname{Jac}(f)\Big|_{(A,a_1,a_2)}$ is nonsingular if and only $\operatorname{Jac}(g)\Big|_{(A,a_1,a_2)}$ is nonsingular. Note that the Jacobian of g evaluated at (A, a_1, a_2) is:

$$\operatorname{Jac}(g)\Big|_{(A,a_1,a_2)} = \left[\begin{array}{ccc|ccc}
I_{i_1j_1} & \cdots & I_{i_{n-1}j_{n-1}} & I_{11} & \cdots & I_{nn} \\
A_{i_1j_1} & \cdots & A_{i_{n-1}j_{n-1}} & A_{11} & \cdots & A_{nn} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
A^{n-1}_{i_1j_1} & \cdots & A^{n-1}_{i_{n-1}j_{n-1}} & A^{n-1}_{11} & \cdots & A^{n-1}_{nn} \\
\hline
\widehat{I}_{i_1j_1} & \cdots & \widehat{I}_{i_{n-1}j_{n-1}} & \widehat{I}_{11} & \cdots & \widehat{I}_{nn} \\
\widehat{A}_{i_1j_1} & \cdots & \widehat{A}_{i_{n-1}j_{n-1}} & \widehat{A}_{11} & \cdots & \widehat{A}_{nn} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\widehat{A}^{n-1}_{i_1j_1} & \cdots & \widehat{A}^{n-1}_{i_{n-1}j_{n-1}} & \widehat{A}^{n-1}_{11} & \cdots & \widehat{A}^{n-1}_{nn}
\end{array}\right]$$

Reordering the rows and the columns of the above matrix we can write it as $\left[\begin{array}{c|c} \operatorname{Jac}_\alpha & \operatorname{Jac}_\beta \end{array}\right]$, where

$$
\mathrm{Jac}_\alpha = \left[\begin{array}{ccc|ccc|c}
I[\alpha]_{11} & \cdots & I[\alpha]_{kk} & I[\alpha]_{i_1 j_1} & \cdots & I[\alpha]_{i_{k-1} j_{k-1}} & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
A[\alpha]_{11}^{k-1} & \cdots & A[\alpha]_{kk}^{k-1} & A[\alpha]_{i_1 j_1}^{k-1} & \cdots & A[\alpha]_{i_{k-1} j_{k-1}}^{k-1} & 0 \\
\hline
A[\alpha]_{11}^{k} & \cdots & A[\alpha]_{kk}^{k} & A[\alpha]_{i_1 j_1}^{k} & \cdots & A[\alpha]_{i_{k-1} j_{k-1}}^{k} & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
A[\alpha]_{11}^{n-1} & \cdots & A[\alpha]_{kk}^{n-1} & A[\alpha]_{i_1 j_1}^{n-1} & \cdots & A[\alpha]_{i_{k-1} j_{k-1}}^{n-1} & 0 \\
\hline
I[\alpha]_{11} & \cdots & I[\alpha]_{kk} & I[\alpha]_{i_1 j_1} & \cdots & I[\alpha]_{i_{k-1} j_{k-1}} & * \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\widehat{A}[\alpha]_{11}^{k-1} & \cdots & \widehat{A}[\alpha]_{kk}^{k-1} & \widehat{A}[\alpha]_{i_1 j_1}^{k-1} & \cdots & \widehat{A}[\alpha]_{i_{k-1} j_{k-1}}^{k-1} & * \\
\hline
\widehat{A}[\alpha]_{11}^{k} & \cdots & \widehat{A}[\alpha]_{kk}^{k} & \widehat{A}[\alpha]_{i_1 j_1}^{k} & \cdots & \widehat{A}[\alpha]_{i_{k-1} j_{k-1}}^{k} & * \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\widehat{A}[\alpha]_{11}^{n-1} & \cdots & \widehat{A}[\alpha]_{kk}^{n-1} & \widehat{A}[\alpha]_{i_1 j_1}^{n-1} & \cdots & \widehat{A}[\alpha]_{i_{k-1} j_{k-1}}^{n-1} & *
\end{array}\right],
$$

and

$$\operatorname{Jac}_\beta = \left[\begin{array}{ccc|ccc|c}
I[\beta]_{kk} & \cdots & I[\beta]_{nn} & I[\beta]_{i_kj_k} & \cdots & I[\beta]_{i_{n-1}j_{n-1}} & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
A[\beta]_{kk}^{k-1} & \cdots & A[\beta]_{nn}^{k-1} & A[\beta]_{i_kj_k}^{k-1} & \cdots & A[\beta]_{i_{n-1}j_{n-1}}^{k-1} & 0 \\
\hline
A[\beta]_{kk}^{k} & \cdots & A[\beta]_{nn}^{k} & A[\beta]_{i_kj_k}^{k} & \cdots & A[\beta]_{i_{n-1}j_{n-1}}^{k} & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
A[\beta]_{kk}^{n-1} & \cdots & A[\beta]_{nn}^{n-1} & A[\beta]_{i_kj_k}^{n-1} & \cdots & A[\beta]_{i_{n-1}j_{n-1}}^{n-1} & 0 \\
\hline
I[\beta]_{kk} & \cdots & I[\beta]_{nn} & I[\beta]_{i_kj_k} & \cdots & I[\beta]_{i_{n-1}j_{n-1}} & * \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\widehat{A}[\beta]_{kk}^{k-1} & \cdots & \widehat{A}[\beta]_{nn}^{k-1} & \widehat{A}[\beta]_{i_kj_k}^{k-1} & \cdots & \widehat{A}[\beta]_{i_{n-1}j_{n-1}}^{k-1} & * \\
\hline
\widehat{A}[\beta]_{kk}^{k} & \cdots & \widehat{A}[\beta]_{nn}^{k} & \widehat{A}[\beta]_{i_kj_k}^{k} & \cdots & \widehat{A}[\beta]_{i_{n-1}j_{n-1}}^{k} & * \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\widehat{A}[\beta]_{kk}^{n-1} & \cdots & \widehat{A}[\beta]_{nn}^{n-1} & \widehat{A}[\beta]_{i_kj_k}^{n-1} & \cdots & \widehat{A}[\beta]_{i_{n-1}j_{n-1}}^{n-1} & *
\end{array}\right],$$

where $\widehat{A} := A + a_1E_{11} + a_2E_{22}$. In $\operatorname{Jac}_\alpha$ the first two block columns represent derivatives with respect to variables in A_1, while the third block column (the last column) represents derivatives with respect to y, and in Jac_β the first two block columns represent derivatives with respect to variables in A_2, and the third block column (the last column) represents derivatives with respect to z. Furthermore, the first two block rows represent derivatives of $\operatorname{tr} M^i$ and the last two block rows represent the derivatives of $\operatorname{tr} N^i$.

Now, suppose

$$\boldsymbol{r}^T \operatorname{Jac}(g)\Big|_{(A,a_1,a_2)} = \boldsymbol{0}^T, \tag{4.20}$$

for some vector $\boldsymbol{r}^T = (\boldsymbol{s}^T, \boldsymbol{t}^T)$, where $\boldsymbol{s}^T = (s_1, \ldots, s_n)$ and $\boldsymbol{t}^T = (t_1, \ldots, t_n)$. Let $s(x) = \sum_{i=1}^n s_ix^i$ and $t(x) = \sum_{i=1}^n t_ix^i$. Then (4.20) holds if and only if $(s(A) + t(\widehat{A})) \circ A = O$ and

$(s(A)+t(\widehat{A}))\circ I = O$, which is equivalent to having

$$(s(A_1)+t(\widehat{A_1}))\circ A_1 = O, (s(A_1)+t(\widehat{A_1}))\circ I = O, \tag{4.21}$$

$$\text{and } (s(A_2)+t(\widehat{A_2}))\circ A_2 = O, (s(A_2)+t(\widehat{A_2}))\circ I = O, \tag{4.22}$$

where I denotes the identity matrix of appropriate size in each case. Let $C_{A_i}(x)$ denote the characterisitc polynomial of A_i for $i=1,2$. Note that by Cayley-Hamilton Theorem $C_{A_i}(A_i) = O$ [56]. For $i=1,2$ let $s_i(x)$ denote the remainder of division of $s(x)$ by the characteristic polynomial of A_i, and t_i denote the remainder of division of $t(x)$ by the characteristic polynomial of $\widehat{A_i}$. Then (4.21) and (4.22) hold if and only if

$$(s_1(A_1)+t_1(\widehat{A_1}))\circ A_1 = O, (s_1(A_1)+t_1(\widehat{A_1}))\circ I = O, \tag{4.23}$$

$$\text{and } (s_2(A_2)+t_2(\widehat{A_2}))\circ A_2 = O, (s_2(A_2)+t_2(\widehat{A_2}))\circ I = O, \tag{4.24}$$

By Theorem 3.2.5 we have $s_i(x)=0$ and $t_i(x)=0$ for $i=1,2$. So, the characteristic polynomials of A_1 and A_2 divide $s(x)$, and the characteristic polynomials of $\widehat{A_1}$ and $\widehat{A_2}$ divide $t(x)$. But since $C_{A_1}(x)$ and $C_{A_2}(x)$ are relatively prime, $C_{A_1}(x)C_{A_2}(x)$ divides $s(x)$. On the other hand, $\deg(C_{A_1}(x)C_{A_2}(x)) = n$ and $\deg(s(x)) = n-1$, hence $s(x)=0$. Similarly, $t(x)=0$, and consequently, $r(x)=0$. This proves that the rows of the $\mathrm{Jac}(g)$ evaluated at (A,a_1,a_2) are linearly independent, and thus $\mathrm{Jac}(f)$ evaluated at (A,a_1,a_2) is nonsingular.

Similar to the proof of Theorem 4.3.4, let $\boldsymbol{a}$ be the assignment of the x_j's corresponding to A, then $g(\boldsymbol{a},a_1,a_2,0,0,\ldots,0) = (\boldsymbol{c},\boldsymbol{d})$. There is an open neighborhood U of $(\boldsymbol{a},a_1,a_2,0,\ldots,0)$ each of whose elements has no zeros in the same $2n+1$ entries. By the Implicit Function Theorem 2.1.1, there is an open neighborhood V of $\boldsymbol{a}$ and an open neighborhood W of $\mathbf{0}$ such that $V\times W\subseteq U$ and for each $\boldsymbol{y}\in W$ there is an $\boldsymbol{x}\in V$ such that $f(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{c},\boldsymbol{d})$. Take $\boldsymbol{y}$ to be a vector in W with no zero entries on the positions corresponding to the edges in G. Then the $(\boldsymbol{x},\boldsymbol{y})$ satisfying $f(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{c},\boldsymbol{d})$ corresponds to a matrix $\widehat{A}\in S(G)$ such that the λ's are the eigenvalues of $\widehat{A}$ and the τ's are the eigenvalues of $\widehat{A}$. $\square$

Chapter 5

The Nowhere-zero Eigenbasis Problem

Motivated by a question asked by Shaun Fallat, we prove that for any n distinct real numbers $\lambda_1 < \lambda_2 < \cdots < \lambda_n$, and for a given connected graph G, there is a real symmetric matrix A such that $\mathcal{G}(A) = G$, the eigenvalues of A are $\lambda_1, \ldots, \lambda_n$, and each entry in each eigenvector of A is nonzero. This is done by first proving the result for when the given graph is a tree, then the results are extended to any connected graph using the solution to the λ-μ problem and continuity.

Note that if the j-th entry of an eigenvector of A is zero then A and $A(j)$ share the eigenvalue corresponding to that eigenvector. For example, assume that

$$\hat{\boldsymbol{v}} = \left[\begin{array}{c} 0 \\ \hline \\ \boldsymbol{v} \\ \\ \end{array}\right]$$

is an eigenvector of A corresponding to the eigenvalue λ. That is, $A\hat{\boldsymbol{v}} = \lambda\hat{\boldsymbol{v}}$. Then $A(1)\boldsymbol{v} = \lambda\boldsymbol{v}$. In order to realize A in a way that none of its eigenvectors have a zero entry, one idea is to choose $n-1$ real numbers μ_j's such that they strictly interlace the λ_j's. Then by Theorem 3.1.1, there is a real symmetric matrix A such that A and $A(1)$ do not share an eigenvalue, hence the first entry of each eigenvector of A is nonzero. However, this is not sufficient to have all the eigenvectors with no zero entries. For example, the graph of the following matrix is a star on 4 vertices

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 0 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 0 & 0 & 4 \end{bmatrix},$$

and its eigenvalues are approximately 2, -0.164, 2.773, and 4.391. We choose to delete vertex 2 (a pendent vertex) and we choose μ's to be approximately 0.186, 2.471, and 4.343 which strictly interlace the spectrum of A. But the eigenvalues of $A(1)$ are $2, 2, 4$. That means the first entry of some of the eigenvectors of A are zero.

Problem 6. ***The λ-SIEP with nowhere-zero eigenbasis.*** *Given a connected graph G on n vertices $1, 2, \ldots, n$ and a set of distinct real numbers $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ find a real symmetric matrix A with spectrum Λ such that graph of A is G, and none of the eigenvectors of A have a zero entry.*

Let us first solve this problem for trees. We will use both results from the λ-μ and the λ-τ-SIEP's for trees. Then we will extend the result to any connected graph. We first mention some preliminary results that we are going to use. A vertex v is a *Parter vertex* of A for the eigenvalue λ if the multiplicity of λ in $A(v)$ is one more than the multiplicity of λ in A. The following is implied by Corollary 3.2 of [57].

Proposition 5.1.7. *Let A be an $n \times n$ singular matrix whose graph is a tree T and let $\boldsymbol{x} = (x_k)$ be a null vector of A. If i and j are adjacent vertices such that $x_i = 0$ and $x_j \neq 0$, then i is a Parter vertex of A.*

Proof. By Corollary 3.2 of [57] the result holds for the zero eigenvalue of $A - \lambda I$. Thus, it holds for the λ eigenvalue of A. $\square$

For the discussion below, fix v and let α_v be the set of vertices w of G such that the path from w to 1 does not pass through v, and β_v be the rest of the vertices, as in Figure

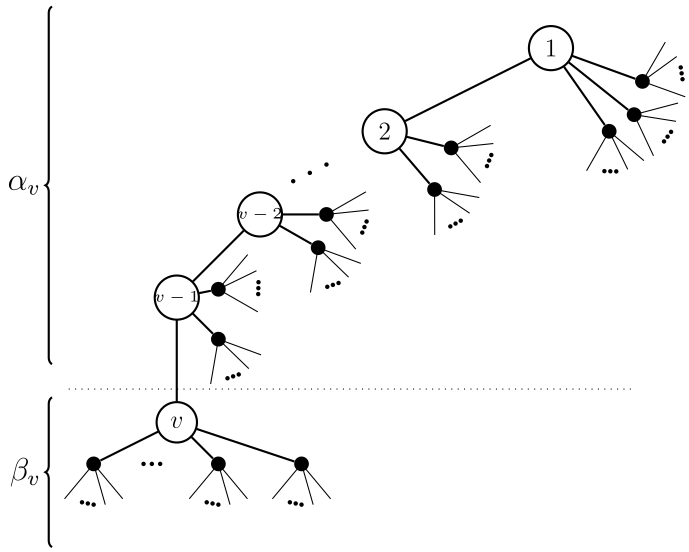


Figure 5.1: A tree T, a fixed vertex v, α_v and β_v.

5.1.

Theorem 5.1.8. *Given a tree G on n vertices $1, 2, \ldots, n$ and a set of distinct real numbers $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ there is a real symmetric matrix A with spectrum Λ such that graph of A is G, and none of the eigenvectors of A has a zero entry.*

Proof. Without loss of generality assume that $\lambda_1 < \lambda_2 < \cdots < \lambda_n$, and 1 is a pendent vertex of G adjacent to 2 (see Figure 5.2).

Choose a set of $n-2$ distinct real numbers $T = \{\tau_1, \tau_2, \ldots, \tau_{n-2}\}$ such that

$$\lambda_1 < \tau_1 < \lambda_2 < \tau_2 < \cdots < \lambda_{n-1} < \tau_{n-2} < \lambda_{n-1} < \lambda_n.$$

Since there is no τ-pairing in the above sequence, Theorem 4.2.1 guarantees a real symmetric matrix A such that $\mathcal{G}(A) = G$, $\sigma(A) = \Lambda$, and $\sigma(A(\{1,2\})) = T$. Note that

$$C_A(x) = (x - a_{11})C_{A(1)}(x) - a_{12}^2 C_{A(\{1,2\})}(x).$$

If $C_A(x)$ and $C_{A(1)}(x)$ share a zero λ, then λ is also a zero of $C_{A(\{1,2\})}$. But since the τ's are distinct from the λ's, $\sigma(A(1)) \cap \sigma(A) = \emptyset$.

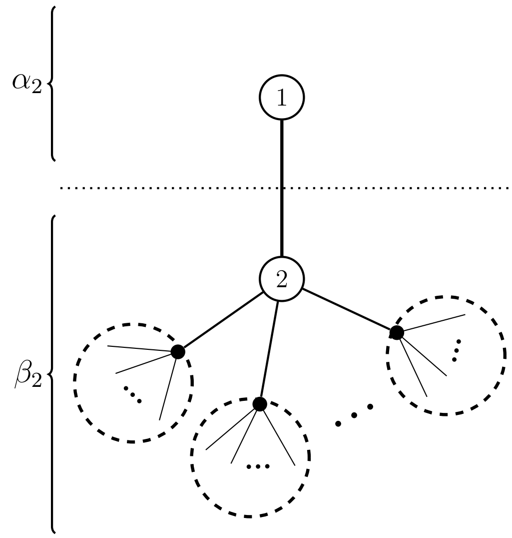


Figure 5.2: Tree G with the pendent vertex 1 adjacent to vertex 2.

Taking $v = 2$, we have

$$C_A(x) = C_{A[\alpha_2]}(x)C_{A[\beta_2]}(x) - a_{12}^2 C_{A[\alpha_2\setminus\{1\}]}(x)C_{A[\beta_2\setminus\{2\}]}(x),$$

where $C_{A[\alpha_2]}(x) = x - a_{11}$ and $C_{A[\alpha_2\setminus\{1\}]}(x) = 1$. If $C_A(x)$ and $C_{A(2)}(x)$ share a zero λ, then $a_{11} = \lambda$, because the τ's are distinct from the λ's. But that is impossible since it implies that λ is also a zero of $C_{A[\alpha_2\setminus\{1\}]}(x)C_{A[\beta_2\setminus\{2\}]}(x) = \prod_{i=1}^{n-2}(x - \tau_i)$. Thus $\sigma(A(2)) \cap \sigma(A) = \emptyset$.

Now, let v be a closest vertex to 1 such that $\sigma(A(v)) \cap \sigma(A) \neq \emptyset$, and without loss of generality assume that $1, 2, \ldots, v-2, v-1, v$ is the path from 1 to v. If there is not such a v then we are done. Note that by above discussion $v \notin \{1, 2\}$. Also, note that

$$C_A(x) = (x - a_{v-2,v-2})C_{A(v-2)}(x) - \sum_{w \in \mathcal{N}(v-2)} a_{v-2,w}^2 C_{A_{w'}(v-2)}(x).$$

If v is a pendent vertex, then

$$C_A(x) = (x - a_{vv})C_{A(v)}(x) - a_{v,v-1}^2 C_{A(\{v,v-1\})}(x).$$

Note that $C_{A(v)}(x)$ does not share a zero with $C_A(x)$, since $C_{A(v-1)}(x)$ does not share any zeros with $C_A(x)$.

Let $B = A[\beta_v]$ and $C = A[\alpha_v]$. Also, let $\Lambda' = \sigma(B)$. Note that there is a set of $|\alpha_v| - 1$ distinct real numbers $M' = \{\mu_1', \mu_2', \ldots, \mu_{|\alpha_v|-1}'\}$ (entry-wise) close to $\sigma(B(v))$ such that M'

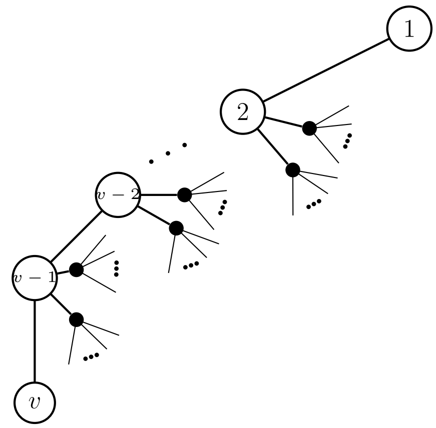


Figure 5.3: Tree G with pendent vertices 1 and v.

strictly interlaces Λ', and $M' \cap \Lambda = \emptyset$, and the polynomial

$$p(x) = \sum_{\substack{w \in \mathcal{N}(v-1) \\ w \neq v}} a_{v-1,w}^2 C_{A_{w'(v-1)}}(x) + a_{v-1,v}^2 \prod_{i=1}^{|\alpha_v|-1} (x - \mu_i')$$

does not have any zeros equal to any λ_i's. (It will be made clear why we need this condition, later.)

Now we are going to reconstruct B so that it doesn't have any common eigenvalues with A. By Theorem 3.1.1 there is a real symmetric matrix B' such that $\mathcal{G}(B') = G[\alpha_v]$, $\sigma(B') = \Lambda'$ and $\sigma(B'(v)) = M'$. Let A' be the matrix obtained form A by replacing B with B'. If $\lambda \in \sigma(C) \cap \Lambda$, then the v-th entry of the eigenvector of A' corresponding to λ is zero, but the $(v-1)$-th entry is nonzero, since v was the closest vertex to 1 with the property that $A(v)$ and A have a common eigenvalue. By Proposition 5.1.7 v is a Parter vertex. That is, the multiplicity of λ in $A'(v)$ is 2, since there is no λ in $B'(v)$. That means by Cauchy Interlacing Inequalities that the multiplicity of λ in $C(v-1)$ is at least 1. But $C(v-1)$ didn't have λ as an eigenvalue, since v was the closest to 1 with a λ as a common eigenvalue. Thus, $\sigma(A'(v)) \cap \Lambda = \emptyset$.

Note that

$$C_A(x) = (x - a_{v-1,v-1}) C_{A(v-1)} - \sum_{w \in \mathcal{N}(v-1)} a_{v-1,w}^2 C_{A_{w'(v-1)}}.$$

When B is replaced by B', the characteristic polynomials $C_A(x)$ and $C_{A_{w'(v-1)}}$ are not changed, for $w \in \mathcal{N}(v-1) \setminus \{v\}$. If $C_A(x)$ and $C_{A(v-1)}(x)$ have a common zero λ, then $\sum_{w \in \mathcal{N}(v-1)} a_{v-1,w}^2 C_{A_{w'}(v-1)}$ has a zero λ, but the μ_i''s are chosen in a way that this sum is not zero when $x = \lambda$.

So far we have shown that the above process of replacing B by B' changes A to an A' such that

- $\sigma(A(v-1)) \cap \sigma(A)$ is still empty, since the perturbation of μ to μ' was small,
- vertex v is no longer a closest vertex to 1 such that $\sigma(A) \cap \sigma(A(v)) \neq \emptyset$, and
- for the vertices in $w \in \alpha_v \setminus \{v-1\}$, we have $\sigma(A(w)) = \sigma(A'(w))$.

Note that after this process there is one fewer vertex w such that $A(w)$ has a common eigenvalue with A and they are not closer than v to 1. Repeat the above process with other closest vertices v' to 1, in order to have $\sigma(A(v')) \cap \sigma(A) \neq \emptyset$. Since the graph is finite, this process ends in finitely many repeats. That is, after finitely many steps $\sigma(A(v)) \cap \sigma(A) \neq \emptyset$, for all vertices v. Consequently, none of the eigenvectors of A have a zero entry. □

Now we can use our proof of Theorem 3.2.5 to extend the above result to any connected graph.

Theorem 5.1.9. *Given a connected graph G on n vertices $1, 2, \ldots, n$ and a set of distinct real numbers $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ there is a real symmetric matrix A with spectrum Λ such that graph of A is G, and none of the eigenvectors of A has a zero entry.*

Proof. Let T be a spanning tree of G. By Theorem 5.1.8 there is a matrix A with spectrum Λ whose graph is T and all of the eigenvectors of A are nowhere-zero. This means that A has the Duarte property with respect to each vertex. Then by Theorem 3.2.5 $\mathrm{Jac}(f)\big|_A$ has full row rank, for f defined by (3.4). Then by Remark 3.3.2 any supergraph G of T can be realized by a matrix $\overline{A}$ with the same spectrum as A, and the spectrum of $\overline{A(v)}$ arbitrarily close to spectrum of $A(v)$, for all v. That is, if an entry of an eigenvector of A is nonzero, it remains nonzero in the corresponding eigenvector of $\overline{A}$. Thus $\bar{A}$ is a matrix with spectrum Λ, graph G, and none of its eigenvectors have a zero entry.

□

Chapter 6

Future Work

In this chapter we state and discuss a variety of related problems that will be considered for future work.

In all of the problems solved in this dissertation we always assumed that the given eigenvalues are distinct. However, many times we are interested in the case when there are some multiple eigenvalues [45, 58–64]. The key idea in the Jacobian method is to use the Implicit Function Theorem when the Jacobian of some certain function at some point is nonsingular. But for the functions that we have considered in this dissertation, the Jacobian evaluated at the matrices of our interest is singular, whenever there is a multiple eigenvalue. So, the Jacobian method in the way that it is presented here fails. However these Jacobian matrices have certain ranks. For example, the Jacobian of the function f defined by (2.2) evaluated at a diagonal matrix $A = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ has rank k, where k is the number of distinct λ_i's. The following example illustrates the singularity of the Jacobian matrix for a small λ-μ-SIEP with multiple eigenvalues.

Example 6.1.10. *Let $G = K_3$ and $i = 1$. We want to construct a 3×3 matrix A with eigenvalues, say 1, 1 and 3 such that the eigenvalues of $A(1)$ are 1 and 2, and $\mathcal{G}(A) = K_3$.*

First, we choose an spanning tree of G and apply Duarte's method on it to realize the given spectral data. Note that according to Duarte [3] we expect some of the entries corre-

Figure 6.1: The graph C_3 and a spanning tree of it.

sponding to some of the edges to be equal to zero, since there are some multiple eigenvalues.

$$A = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix}.$$

Constructing the matrix M and function f and evaluating the Jacobian of f at A, similar to Example 3.3.3, up to some scaling of the rows we get

$$\mathrm{Jac}_x(f)\Big|_A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 4 & 2 & 4 & 0 & 1 \\ 17 & 4 & 17 & 0 & 8 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 2 & 4 & 0 & 0 \end{bmatrix}.$$

Then rank$(\mathrm{Jac}_x(f)\big|_A) = 4$.

If one can show that this rank is constant in some (probably lower dimensional) neighborhood of A, this suggests using the 'Constant Rank Theorem' [47](stated below), instead of the Implicit Function Theorem.

Theorem 6.1.11 (Constant Rank Theorem)**.** *Suppose $U \subseteq \mathbb{R}^n$, $F = (f_1, \ldots, f_m) : U \to \mathbb{R}^m$ is continuously differentiable infinitely many times in a neighborhood of a, and rank*$(\mathrm{Jac}(F)\big|_x) =$ *k for all x in a neighborhood of a. Then there are open neighborhoods V of a and W of $F(a)$ and diffeomorphisms $\phi : V \to \mathbb{R}^n$ and $\psi : W \to \mathbb{R}^m$ with*

$$\begin{tikzcd} V \arrow[r, "F"] \arrow[d, "\phi"'] & W \arrow[d, "\psi"] \\ \mathbb{R}^n \arrow[r] & \mathbb{R}^m \end{tikzcd}$$

such that $\psi \circ F \circ \phi^{-1}(x_1, \ldots, x_n) = (x_1, \ldots, x_k, 0, \ldots, 0)$.

It is clear that if the rank of $\mathrm{Jac}(F)$ at a is k, then it is at least k in a neighborhood of a. One main concern here would be to find a neighborhood of a such that the rank of the Jacobian matrix does not increase. This suggests a modified version of the Jacobian method that would allow us to realize 'some' superpatterns of a matrix rather than 'all' of them.

So, one might approach the structured inverse eigenvalue problems solved in this dissertation when the eigenvalues are not distinct.

Problem 7. ***The general λ-SIEP for graphs:*** *A multi-set of real numbers*

$$\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$$

and a graph G on n vertices $1, 2, \ldots, n$ are given. Find a real symmetric matrix A such that $\mathcal{G}(A) = G$, and $\sigma(A) = \Lambda$.

Note that in particular solving Problem 7 also solves the general λ-SIEP for trees, which would be a major result. Furthermore, this suggests that we consider the λ-μ-SIEP with a broader perspective. While the problem can be stated as one general problem, here we state it as two problems. First problem concerns distinct λ's and distinct μ's, but allowing coincidences between the λ's and μ's. Second problem allows multiplicities in λ's and μ's, but no strict interlacing inequalities, except those that are forced by multiplicities. The results by Barrett et al. [48,49], Johnson et al. [42,60,65], and Oblaka et al. [66] suggest that these might be different questions in nature.

Problem 8. ***The first general λ-μ-SIEP for graphs:*** *Two sets of distinct real numbers*

$$\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}, M = \{\mu_1, \mu_2, \ldots, \mu_{n-1}\}$$

and a graph G on n vertices $1, 2, \ldots, n$ are given, such that M interlaces Λ. For a fixed $w \in \{1, 2, \ldots, n\}$ does there exist a real symmetric matrix A such that $\mathcal{G}(A) = G$, $\sigma(A) = \Lambda$, and $\sigma(A(w)) = M$? More precisely, characterize all graphs G such that there is a real symmetric matrix A with $\mathcal{G}(A) = G$, $\sigma(A) = \Lambda$, and $\sigma(A(w)) = M$.

Problem 9. ***The second general λ-μ-SIEP for graphs:*** *Two multi-sets of real numbers*

$$\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}, M = \{\mu_1, \mu_2, \ldots, \mu_{n-1}\}$$

and a graph G on n vertices $1, 2, \ldots, n$ are given. Let Λ' and M' be the sets obtained from Λ and $M \setminus \Lambda$ by keeping only distinct numbers, and assume that M' strictly interlaces Λ'. For a fixed $w \in \{1, 2, \ldots, n\}$ does there exist a real symmetric matrix A such that $\mathcal{G}(A) = G$, $\sigma(A) = \Lambda$, and $\sigma(A(w)) = M$? More precisely, characterize all graphs G such that there is a real symmetric matrix A with $\mathcal{G}(A) = G$, $\sigma(A) = \Lambda$, and $\sigma(A(w)) = M$.

Similar questions can be asked as analogues of the λ-τ-SIEP, and also the problems of perturbing some diagonal entries. Here we mention them in their most general form.

Problem 10. ***The general λ-τ-SIEP for graphs:*** *Two multi-sets of real numbers*

$$\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}, T = \{\tau_1, \tau_2, \ldots, \tau_{n-2}\}$$

and a graph G on n vertices $1, 2, \ldots, n$ are given such that T and Λ satisfy the second order Cauchy interlacing inequalities. For fixed r and s in $\{1, 2, \ldots, n\}$ does there exist a real symmetric matrix A such that $\mathcal{G}(A) = G$, $\sigma(A) = \Lambda$, and $\sigma(A(\{r, s\})) = T$? More precisely, characterize all graphs G such that there is a real symmetric matrix A with $\mathcal{G}(A) = G$, $\sigma(A) = \Lambda$, and $\sigma(A(\{r, s\})) = M$.

While we have considered questions about the existence of a matrix of size n with a given graph and spectrum, and the spectrum of a certain submatrix of size $n-1$ or $n-2$, the question when the submatrix is of size $n-k$, for $k \geq 3$ remains wide-open.

Note that for $W \subseteq \{1, 2, \ldots, n\}$ of size k, the eigenvalues of $A(W)$ and the eigenvalues of A should satisfy the k-th order Cauchy interlacing inequalities. The following lemma can be proved easily by inducting on k and using first order Cauchy interlacing inequalities.

Lemma 6.1.12 (General Cauchy interlacing inequalities)**.** *Let A be an $n \times n$ real symmetric matrix with eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, and let B be an $(n-k) \times (n-k)$ principal submatrix of A with eigenvalues $\gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_{n-k}$. Then*

$$\lambda_i \leq \gamma_i \leq \lambda_{i+k}, \qquad i = 1, 2, \ldots, n-k \tag{6.1}$$

Problem 11. ***The general λ-γ-SIEP for graphs:*** *Two multi-sets of real numbers*

$$\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}, \Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_{n-k}\}$$

and a graph G on n vertices $1, 2, \ldots, n$ are given such that Γ and Λ satisfy the k-th order Cauchy interlacing inequalities. For fixed $W \subseteq \{1, 2, \ldots, n\}$ of size k does there exist a real symmetric matrix A such that $\mathcal{G}(A) = G$, $\sigma(A) = \Lambda$, and $\sigma(A(W)) = \Gamma$? More precisely, characterize all graphs G such that there is a real symmetric matrix A with $\mathcal{G}(A) = G$, $\sigma(A) = \Lambda$, and $\sigma(A(W)) = \Gamma$.

So far we have considered several problems in which one removes some rows and columns of a matrix A to obtain B and prescribing the eigenvalues of A and B and the graph of A. However, this is not the only way of approaching such problems.

Problem 12. *Let G be a graph on n vertices and $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be n real numbers. Is there a real symmetric matrix A such that $\mathcal{G}(A) = G$ and $\lambda_k \in \sigma(A[1, 2, \ldots, k])$, for $k = 1, 2, \ldots, n$?*

Here is a rough idea for the case that G is the complete graph on n vertices, and λ_i's are distinct. Let A_k denote the principal submatrix $A[1, 2, \ldots, k]$ for $k = 1, \ldots, n$. Define a function f on the set of real symmetric matrices $f : \mathbb{R}^{\frac{n(n+1)}{2}} \to \mathbb{R}^{\frac{n(n+1)}{2}}$ by

$$f : A \mapsto \left(\mathrm{tr}(A_1), \mathrm{tr}(A_2), \mathrm{tr}(A_2^2), \ldots, \mathrm{tr}(A_k), \mathrm{tr}(A_k^2), \mathrm{tr}(A_k^k), \ldots, \mathrm{tr}(A_n), \mathrm{tr}(A_n^2), \mathrm{tr}(A_n^n)\right).$$

Show that $\mathrm{Jac}(f)\big|_{\mathrm{diag}(\lambda_1, \ldots, \lambda_n)}$ is nonsingular, and proceed by the Jacobian method.

Furthermore, one can consider other families of matrices. For example, Sudipta Mallik and the author have studied the λ-μ-SIEP for the family of skew-symmetric matrices using the Jacobian method in [67]. Keep in mind that while the eigenvalues of real skew-symmetric matrices are purely imaginary numbers and they come in conjugate pairs, they still satisfy interlacing inequalities on the imaginary axis of the complex plane. So, all the above questions can be asked for appropriate choices of sets Λ, M, T, and Γ. All such questions can be asked about larger families of matrices; for example, how could the results in this dissertation be generalized if the matrix A is not required to be symmetric, or real.

Moreover, the zero-nonzero pattern of a matrix is only one of the structures of matrices of interest in science and engineering. What are the questions that can be asked about other structures, such as Hessenberg matrices, circulant and Toeplitz matrices, Hankel matrices, totally nonnegative matrices [68] etc.?

Recall that in Theorem 4.3.4 our methods require the existence of a certain type of spanning tree for the graph in order to work. One remaining case is to investigate what will happen if the graph does not have such a spanning tree.

Problem 13. *Let* $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ *and* $T = \{\tau_1, \ldots, \tau_{n-2}\}$ *form a nondegenerate* λ-τ *sequence with* k τ*-pairings. Let* G *be a connected graph on* n *vertices and* r *and* s *be two adjacent vertices in* G*. Furthermore assume that for any spanning tree of* G *containing the edge* $\{r, s\}$*, either* V_r *or* V_s *has at most* k *vertices. Does there exist a real symmetric matrix* A *such that* $\mathcal{G}(A) = G$*,* $\sigma(A) = \Lambda$*, and* $\sigma(A(\{r, s\})) = T$*?*

Figure 6.2 illustrates an example of such a graph for a nondegenerate λ-τ sequence with a τ pairing:

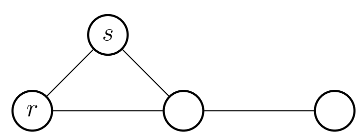


Figure 6.2: An example of a connected graph G on 4 vertices such that no spanning tree of G including the edge $\{r, s\}$ has both $|V_r|, |V_s| > 1$.

A similar question can be asked when r and s are not adjacent in G.

Problem 14. *Let* $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ *and* $T = \{\tau_1, \ldots, \tau_{n-2}\}$ *form a nondegenerate* λ-τ *sequence with* k τ*-pairings. Let* G *be a connected graph on* n *vertices and* r *and* s *be two non-adjacent vertices in* G*. Furthermore, assume that for any spanning tree* T *of* G *Assumption*

1 *does not hold. Does there exist a real symmetric matrix A such that $\mathcal{G}(A) = G$, $\sigma(A) = \Lambda$, and $\sigma(A(\{r, s\})) = T$?*

Figure 6.3 illustrates an example of such a graph for a nondegenerate λ-τ sequence with two τ-pairings:

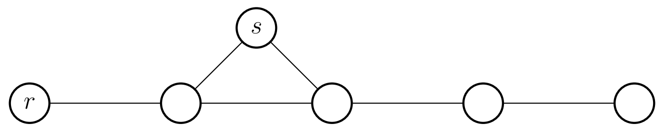


Figure 6.3: An example of a connected graph G on 6 vertices such that no spanning tree of G satisfies Assumption 1.

All in all, it is shown that the Jacobian method is a powerful tool to solve various structured inverse eigenvalue problems, but it also can be used in other various settings such as problems concerning the existence of spectrally arbitrary patterns [69], orthogonal matrices with prescribed sign patterns [70, 71] etc., and many more to be discovered.

Appendix A

SAGE Code Written by the author

A.1 The `lambda_siep()` function

The following SAGE code defines a function `lambda_siep()`, where G is a tree on n vertices and L is a list of n distinct real numbers. The output of the function is an $n \times n$ real symmetric matrix A such that $\mathcal{G}(A) = G$ and $\sigma(A) = L$.

```
# Build variables, and the matrix corresponding to it
def build_variables(n):
    names = [ [[] for i in range(n)] for j in range(n) ]
    for i in range(n):
        for j in range(i+1):
            names[i][j] = (SR('x_' + str(j) + '_' + str(i)))
            names[j][i] = (SR('x_' + str(j) + '_' + str(i)))
    return(names)
###########################################################
# Define the function f that maps a matrix to the coefficients of
[+]its characteristic polynomial
```

```
def CharPoly(Mat):
    X = matrix(Mat)
    n = X.ncols()
    C_X = X.characteristic_polynomial()
    Y = []
    for i in range(n):
        Y.append(C_X[i])
    return(Y)
###########################################################
# This solves that lambda SIEP
def lambda_siep(G,L,iter=100,epsilon = .1):
# G is any graph on n vertices
# L is the list of n desired distinct eigenvalues
# m is the number of itterations of the Newton's method
# epsilon: the off-diagonal entries will be equal to epsilon
    n = G.order()
    my_variables = build_variables(n)
    R = PolynomialRing(CC,[my_variables[i][j] for i in range(n) for
    [+]j in range(n)])
    R.gens()
    R.inject_variables()
    X = [ [ R.gens()[n*i+j] for i in range(n) ] for j in range(n) ]
    Y = matrix(CharPoly(X)) - matrix(CharPoly(diagonal_matrix(L)))
    J = matrix(R,n)
    for i in range(n):
        for j in range(n):
            J[i,j] = derivative(Y[0][i],my_variables[j][j])
    B = diagonal_matrix(L) + epsilon * G.adjacency_matrix()
    count = 0
    while count < iter:
        T = [ B[i,j] for i in range(n) for j in range(n)]
```

```
        C = (J(T)).solve_right(Y(T).transpose())
        LC = list(C)
        B = B - diagonal_matrix([LC[i][0] for i in range(n)])
        count = count + 1
    return(B)
###########################################################
# This shows the output matrix, its eigenvalues and the eigenvlaues
[+]of A(i), and its graph
def check_output_lambda_siep(A,precision=8):
# A is a matrix which is the output of lambda_mu_for_trees()
# i is the one that also is entered in lambda_mu_for_trees()
# precision is an integer that shows how many digits do I want to be
[+] printed at the end, and I set the default to be 8
    eigA = A.eigenvalues()
    EigA = []
    for e in eigA:
        EigA = EigA + [e.n(precision)]
    print('A is:')
    print(A.n(precision))
    print(' ')
    print('Eigenvalues of A are: %s') %(EigA)
    AdjA = matrix(A.ncols())
    for i in range(A.ncols()):
        for j in range(A.ncols()):
            if i != j:
                if A[i,j] != 0:
                    AdjA[i,j] = 1
    FinalGraph = Graph(AdjA)
    print(' ' )
    print('And the graph of A is:')
    FinalGraph.show()
```

```
##########################################################
# Here is a sample input
G = graphs.CompleteGraph(5)
L = [1,2,3,4,5]
A = lambda_siep(G, L, iter=1000, epsilon=.1)
check_output_lambda_siep(A,precision=64)
```

Code A.1: The `lambda_siep()` function.

A.2 The `lambda_mu_for_trees()` function

The following SAGE code defines a function `lambda_mu_for_trees()`, where G is a tree on n vertices, L is a list of n distinct real numbers, M is a list of $n-1$ distinct real numbers that strictly interlaces L, and i is an integer between 0 and $n-1$. The output of the function is an $n \times n$ real symmetric matrix A such that $\mathcal{G}(A) = G$, $\sigma(A) = L$, and $\sigma(A(i)) = M$.

```
##########################################################
# This checks to see if the output is correct, and if it is
# not it prints a meaningful error message.
# This is a boolean function with outputs True and False
def is_input_for_lambda_mu(G,L,M,i):
# G is a tree on n vertices with vertex i
# L is a list of n distinct real numbers
# M is a list of n-1 distinct real numbers that strictly
# interlaces L
# this function checks to so if the above are actually true
    Error = 0
  # this simply indicates which error message to show
    Errors = ["Calculating...", # 0, nothing is wrong
             "There are duplicate lambda's!", # 1
             "There are duplicate mu's!", # 2
```

```
             "Some lambda is equal to some mu!", # 3
             "Number of lambda's should be exactly one more than the
             [+] number of mu's!", # 4
             "'i' should be a vertex of the graph , that is, an
             [+]integer between 0 and %s, but it is %s." %(len(L)-1,
             [+]i), #5
             "G should be either a tree or the adjacency matrix of a
             [+] tree!", # 6
             "Number of eigenvalues should be equal to the number of
             [+] the vertices", # 7
             "Interlacing inequalities are not met!", # 8
             ]
    if len(L) > len(set(L)):
  # Are lambda's distinct?
        Error = 1
    elif len(M) > len(set(M)):
  # Are mu's distinct?
        Error = 2
    elif len(L+M) > len(set(L+M)):
  # Are lambda's different from mu's?
        Error = 3
    elif len(L) != len(M) + 1:
  # Are lambda's one more than mu's?
        Error = 4
    elif not G.has_vertex(i):
  # Is i a vertex of G?
        Error = 5
    elif not G.is_tree():
        Error = 6
    elif len(L) != G.size()+1:
        Error = 7
```

```
    else:
  # Are the Cauchy interlacing inequalities met?
        L.sort()
        M.sort()
        for l in range(len(L)-1):
            if M[l] < L[l] or M[l] > L[l+1]:
                Error = 8
    if Error != 0:
  # Return the error message, if any.
        print Errors[Error]
        print ('--------------------------------------')
        print ('--------------No output!--------------')
        print ('--------------------------------------')
        return(False)
    else:
        return(True)
##########################################################
# This gets two lists, adds the elements of each list, and
# returns the difference of the two sums
def difference_of_sums(L,M):
# L is list of numbers
# M is list of numbers
    sumL = 0
    sumM = 0
    for l in range(len(L)): # Add up lambda's
        sumL = sumL + L[l]
    for m in range(len(M)): # Add up mu's
        sumM = sumM + M[m]
    return(sumL - sumM)
##########################################################
# This finds the connected components of G after deleting
```

```
# vertex i and returns I =[[H_j, e_j] for j ], where H_j are
# connected components of G(i) and e_j is the vertex of H_j
# which was a neighbors of i in G.
def list_of_connected_components(G,i):
# G is a tree with a vertex i
    # Make a copy of G and delete the i-th vertex of it
  # and call it H
    H = copy(G)
    H.delete_vertices([i])
    CCH = H.connected_components()
    N = G.neighbors(i)
  # Get a list of nbrs of vertex i in G
    I = []
  # a list of connected components of [G(i),v] where v is
  # the neighbor of i in G and in that component
    for cc in range(len(CCH)):
        e = (set(CCH[cc])).intersection(set(N)).pop()
    # this intersection has only one element and I want
    # to look at that one element.
        I = I + [[H.subgraph(CCH[cc]),e]]
    return(I)
###########################################################
# This looks lists all the numerators of the partial
# fraction decompositions of prod(x - l_i) / prod(x - m_i)
def numerators_of_pfd(L,M):
    B = []
  # f/g = (x - a) - sum b/(x - u) and B is a list of
  # all b's
    for m in range(len(M)):
        top = 1
        bottom = 1
```

```
        for j in range(len(L)):
            top = top * (M[m] - L[j])
        for j in range(len(M)):
            if j != m:
                bottom = bottom * (M[m] - M[j])
        B = B + [- top / bottom]
    return(B)
###########################################################
# This gets two numbers and two lists and returns the square
# root of the coefficient of y_j along with some other
# things that will be used in further calculations
###########################################################
def coefficient_of_yj(count,s,M,B):
    tempL = []
  # mu's will be broken into lists of size s and each list
  # will be assigned as new lamba's
    tempF = 0
  # This is a rational function that adds enough
  # b / (x - mu) for each component
    for l in range(count, count + s):
  # goes through this connected component
        tempL = tempL + [M[l]]
        tempF = tempF + B[l] / (x - M[l])
    v = ((tempF.numerator()).expand()).leading_coefficient(x)
    w = ((tempF.denominator()).expand()).leading_coefficient(x)
    return(sqrt(v/w),tempF,tempL)
###########################################################
# This piece solves the lambda-mu SIEP for trees
# This is a version of lambda_mu_for_tree() that looks at
# the indices of the graph and if the largest index is n
# (counted from zero), then it returns a (n+1)x(n+1) matrix
```

```
# and the indices that are missing in the graph will have
# zero rows and columns at the end. This is because later I
# want to add the missing vertices from another graph.
def lambda_mu_for_tree(G,L,M,i,Out=None):
# G is a graph or the adjacency matrix of a graph. In case
# that it is a matrix, it's type should match
# 'sage.matrix.matrix_integer_dense.Matrix_integer_dense'
# L is a litst of distinct real numbers to be realized as
# eigenvalues of A
# M is a list of distinct real numbers to be realiezed as
# eigenvalues of A(i)
# i is the vertex to be deleted
    if type(G) == sage.matrix.matrix_integer_dense.
    [+]Matrix_integer_dense:
        G = Graph(G)
    indices = list(G)
    order = max(indices)+1
    if Out is None:
        Out = [[0 for j in range(order)] for j in range(order)]
    if is_input_for_lambda_mu(G,L,M,i): # check the inputs
# Here I want to evaluate the diagonal entry (i,i)
        if len(L) == 1:
    # If there is only one lambda, then that's the
    # diagonal entry
            a = copy(L[0])
            Out[i][i] = copy(a)
            Out = Matrix(RR,Out)
      # I'm not deleting extra rows and columns since
      # I'll need them in lambda_tau problem
            return(Out)
        # If there are more than one lambda, then the
```

```
    # diagonal entry is the difference of traces (sum of
    # lambda's minus sum of mu's)
      a = difference_of_sums(L,M)
      Out[i][i] = copy(a)
      I = list_of_connected_components(G,i)
      B = numerators_of_pfd(L,M)
      count = 0
      for component in range(len(I)):
    # remember 'I' was a list of connected components
        s = I[component][0].size() + 1
     # +1 because size() starts from 0
        e = I[component][1]
        # this gets s, M, B and returns the sqrt(v/w)
        (a,tempF,tempL) = coefficient_of_yj(count,s,M,B)
        # Write the i,j_i entry of the output matrix
        Out[i][e] = Out[e][i] = a
     # this is A_{i, j_i} entry
        # Here I find the roots of the numerator of the
     # sums of b / (x - mu)
        if (tempF.numerator()).degree(x) == 0:
          tempM = []
        else:
          # we know all the solution are real, but
      # there are calculation errors, so we'll
      # find all the complex solutions and then
      # omit the very small imaginary parts
          S = (tempF.numerator()).roots(ring = CC,
          [+]multiplicities = False)
          for solution in range(len(S)):
            S[solution] = S[solution].real()
          tempM = copy(S)
```

```
      count = count + s
      # Here I call lambda_mu_for_tree() for subtree
   # 'component'
      lambda_mu_for_tree(I[component][0], tempL, tempM, e, Out
      [+])
    Out = Matrix(RR,Out)
  # I'm not deleting extra rows and columns since I'll
  # need them in lambda_tau problem
    return(Out)
#############################################################
# This shows the output matrix, its eigenvalues and the
# eigenvlaues of A(i), and its graph
def check_output_lambda_mu_for_tree(A,i,precision=8):
# A is a matrix which is the output of lambda_mu_for_trees()
# i is the one that also is entered in lambda_mu_for_trees()
# precision is an integer that shows how many digits do I
# want to be printed at the end. I set the default to be 8
  eigA = A.eigenvalues()
  EigA = []
  for e in eigA:
    EigA = EigA + [e.n(precision)]
  eigB = (A.delete_rows([i]).delete_columns([i])).eigenvalues()
  EigB = []
  for e in eigB:
    EigB = EigB + [e.n(precision)]
  print('A is:')
  print(A.n(precision))
  print(' ')
  print('Eigenvalues of A are: %s') %(EigA)
  print(' ')
  print('Eigenvalues of A(%s) are: %s') %(i,EigB)
```

```
    AdjA = matrix(A.ncols())
    for i in range(A.ncols()):
        for j in range(A.ncols()):
            if i != j:
                if A[i,j] != 0:
                    AdjA[i,j] = 1
    FinalGraph = Graph(AdjA)
    print(' ' )
    print('And the graph of A is:')
    FinalGraph.show()
###########################################################
# Here is a sample input
G = Graph(graphs.PetersenGraph().min_spanning_tree())
L = [-3,-1,2,4,6,8,11,15,19,23]
M = [-2,1,3,5,7,10,12,17,21]
i = 1
A=lambda_mu_for_tree(G,L,M,i)
check_output_lambda_mu_for_tree(A,i,precision=12)
```

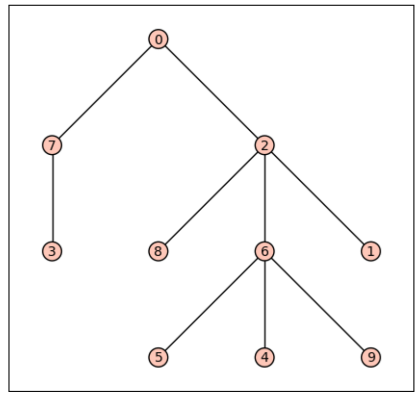
Code A.2: The `lambda_mu_for_trees()` function.

A.2.1 Some outputs

Below are a few test runs of the above algorithm on different trees on 10 vertices. Let $L = [-10, -8, -5, -1, 2, 5, 7, 10, 12, 15]$, and $M = [-9, -6, -3, 1, 3, 6, 9, 11, 13]$.

1. G is the following tree, and $i = 0$.

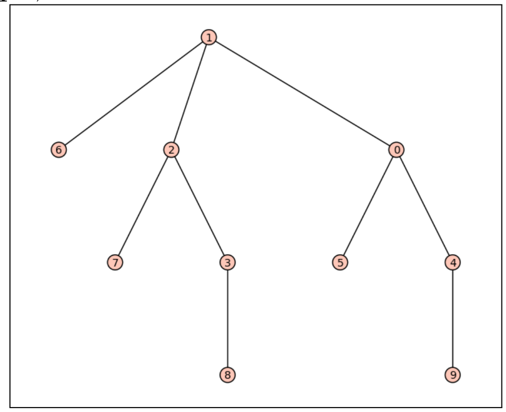
```
timeit('lambda_mu_for_tree(G,L,M,i)')
```




```
5 loops, best of 3: 394 ms per loop
```

$$A \approx \left[\begin{array}{c|ccccccccc}
2.0 & 0 & 8.8 & 0 & 0 & 0 & 0 & 4.2 & 0 & 0 \\
\hline
0 & 5.4 & 2.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
8.8 & 2.0 & 0.91 & 0 & 0 & 0 & 3.9 & 0 & 2.5 & 0 \\
0 & 0 & 0 & 12 & 0 & 0 & 0 & 1.0 & 0 & 0 \\
0 & 0 & 0 & 0 & -4.1 & 0 & 2.2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.4 & 1.5 & 0 & 0 & 0 \\
0 & 0 & 3.9 & 0 & 2.2 & 1.5 & -2.5 & 0 & 0 & 1.9 \\
4.2 & 0 & 0 & 1.0 & 0 & 0 & 0 & 12 & 0 & 0 \\
0 & 0 & 2.5 & 0 & 0 & 0 & 0 & 0 & 7.9 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.9 & 0 & 0 & -7.9
\end{array}\right].$$

2. G is the following spanning tree of the Petersen graph, and $i = 1$.



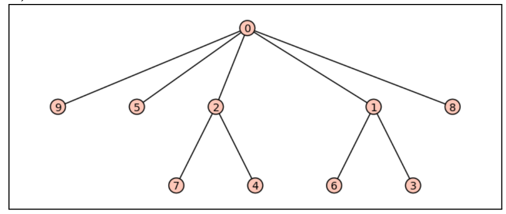

```
timeit('lambda_mu_for_tree(G,L,M,i)')
```

```
5 loops, best of 3: 179 ms per loop
```

$$A \approx \left[\begin{array}{r|r|rrrrrrrr}
-2.6 & 6.7 & 0 & 0 & 2.1 & 2.2 & 0 & 0 & 0 & 0 \\ \hline
6.7 & 2.0 & 6.4 & 0 & 0 & 0 & 2.9 & 0 & 0 & 0 \\ \hline
0 & 6.4 & 6.8 & 3.0 & 0 & 0 & 0 & 1.3 & 0 & 0 \\
0 & 0 & 3.0 & 5.9 & 0 & 0 & 0 & 0 & 1.2 & 0 \\
2.1 & 0 & 0 & 0 & -6.6 & 0 & 0 & 0 & 0 & 1.7 \\
2.2 & 0 & 0 & 0 & 0 & -0.54 & 0 & 0 & 0 & 0 \\
0 & 2.9 & 0 & 0 & 0 & 0 & 13 & 0 & 0 & 0 \\
0 & 0 & 1.3 & 0 & 0 & 0 & 0 & 10 & 0 & 0 \\
0 & 0 & 0 & 1.2 & 0 & 0 & 0 & 0 & 5.9 & 0 \\
0 & 0 & 0 & 0 & 1.7 & 0 & 0 & 0 & 0 & -7.2
\end{array}\right].$$

3. G is the following spanning tree of the Harrary $(5, 10)$ graph, and $i = 0$.

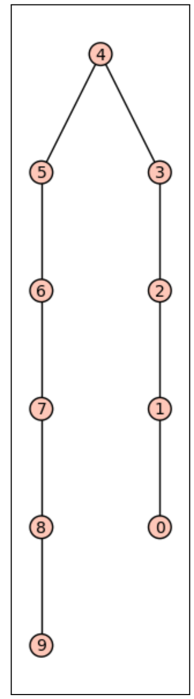
```
timeit('lambda_mu_for_tree(G,L,M,i)')
```




```
5 loops, best of 3: 90.7 ms per loop
```

$$A \approx \left[\begin{array}{c|ccccccccc}
2.0 & 5.5 & 5.9 & 0 & 0 & 3.5 & 0 & 0 & 3.0 & 2.9 \\
\hline
5.5 & -4.4 & 0 & 1.3 & 0 & 0 & 1.5 & 0 & 0 & 0 \\
5.9 & 0 & 2.7 & 0 & 1.1 & 0 & 0 & 1.4 & 0 & 0 \\
0 & 1.3 & 0 & -8.6 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.1 & 0 & 1.9 & 0 & 0 & 0 & 0 & 0 \\
3.5 & 0 & 0 & 0 & 0 & 9.0 & 0 & 0 & 0 & 0 \\
0 & 1.5 & 0 & 0 & 0 & 0 & -5.0 & 0 & 0 & 0 \\
0 & 0 & 1.4 & 0 & 0 & 0 & 0 & 5.4 & 0 & 0 \\
3.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 11 & 0 \\
2.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 13
\end{array}\right].$$

4. G is the following path, and $i = 4$.

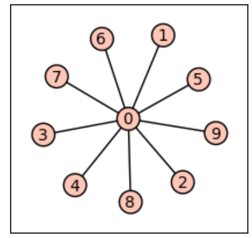



```
timeit('lambda_mu_for_tree(G,L,M,i)')
```

```
5 loops, best of 3: 548 ms per loop
```

$$A \approx \left[\begin{array}{rrrr|r|rrrrr}
10 & 1.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.4 & 9.3 & 2.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2.4 & 9.3 & 2.3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2.3 & 9.9 & 5.9 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 5.9 & 2.0 & 7.7 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 7.7 & -1.2 & 3.6 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 3.6 & -2.7 & 3.4 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 3.4 & -4.2 & 3.2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.2 & -2.4 & 3.3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.3 & -3.5
\end{array}\right].$$

5. G is the following star, and $i = 0$.

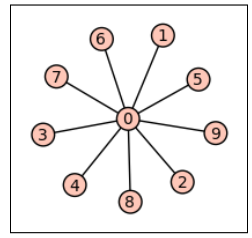



```
timeit('lambda_mu_for_tree(G,L,M,i)')
```

```
25 loops, best of 3: 11.4 ms per loop
```

$$A \approx \left[\begin{array}{c|ccccccccc}
2.0 & 1.7 & 2.9 & 4.3 & 3.8 & 3.8 & 2.4 & 3.5 & 3.0 & 2.9 \\
\hline
1.7 & -9.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2.9 & 0 & -6.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4.3 & 0 & 0 & -3.0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3.8 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\
3.8 & 0 & 0 & 0 & 0 & 3.0 & 0 & 0 & 0 & 0 \\
2.4 & 0 & 0 & 0 & 0 & 0 & 6.0 & 0 & 0 & 0 \\
3.5 & 0 & 0 & 0 & 0 & 0 & 0 & 9.0 & 0 & 0 \\
3.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 11 & 0 \\
2.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 13
\end{array}\right].$$

6. G is the following star, and $i = 1$.




```
timeit('lambda_mu_for_tree(G,L,M,i)')
```

```
5 loops, best of 3: 140 ms per loop
```

$$A \approx \left[\begin{array}{c|c|cccccccc}
2.9 & 9.8 & 1.5 & 1.9 & 3.0 & 1.7 & 2.5 & 3.0 & 1.8 & 1.7 \\
\hline
9.8 & 2.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
1.5 & 0 & -8.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.9 & 0 & 0 & -5.4 & 0 & 0 & 0 & 0 & 0 & 0 \\
3.0 & 0 & 0 & 0 & -1.3 & 0 & 0 & 0 & 0 & 0 \\
1.7 & 0 & 0 & 0 & 0 & 2.0 & 0 & 0 & 0 & 0 \\
2.5 & 0 & 0 & 0 & 0 & 0 & 5.2 & 0 & 0 & 0 \\
3.0 & 0 & 0 & 0 & 0 & 0 & 0 & 7.5 & 0 & 0 \\
1.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10. & 0 \\
1.7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12
\end{array}\right].$$

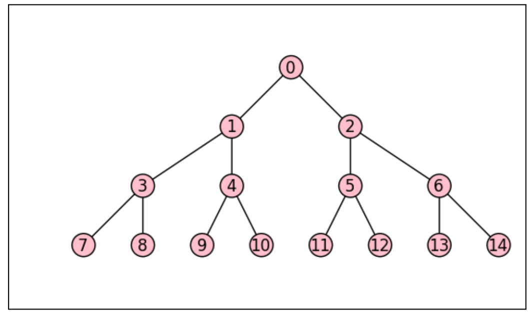
For the following example, let

$$L = [-15, -12, -10, -8, -5, -1, 2, 5, 7, 10, 12, 15, 17, 20, 22]\,,$$

and

$$M = [-13, -11, -9, -6, -3, 1, 3, 6, 9, 11, 13, 16, 18, 21]\,.$$

7. G is the following full binary tree, and $i = 0$.



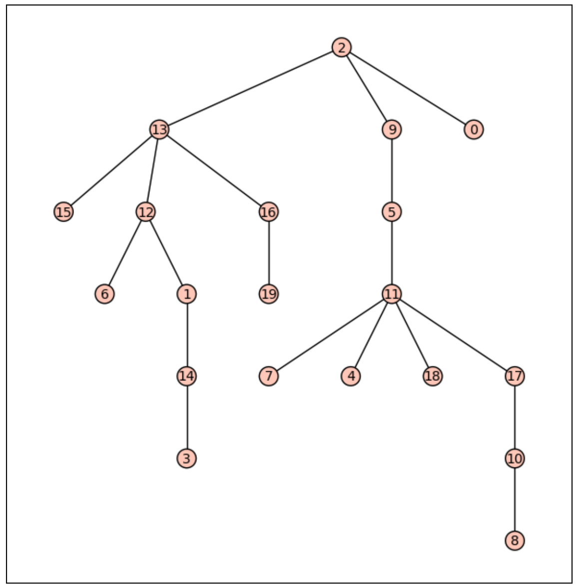

```
timeit('lambda_mu_for_tree(G,L,M,i)')
```

```
5 loops, best of 3: 305 ms per loop
```

$$A \approx \left[\begin{array}{c|cccccccccccccc}
3.0 & 11 & 9.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
11 & -3.1 & 0 & 3.5 & 3.7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
9.6 & 0 & 12 & 0 & 0 & 2.6 & 3.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3.5 & 0 & -10 & 0 & 0 & 0 & 1.2 & 1.2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3.7 & 0 & 0 & -1.6 & 0 & 0 & 0 & 0 & 2.2 & 1.3 & 0 & 0 & 0 & 0 \\
0 & 0 & 2.6 & 0 & 0 & 9.4 & 0 & 0 & 0 & 0 & 0 & 1.9 & 1.3 & 0 & 0 \\
0 & 0 & 3.0 & 0 & 0 & 0 & 17 & 0 & 0 & 0 & 0 & 0 & 0 & 1.1 & 1.8 \\
0 & 0 & 0 & 1.2 & 0 & 0 & 0 & -9.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1.2 & 0 & 0 & 0 & 0 & -12. & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2.2 & 0 & 0 & 0 & 0 & -3.6 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1.3 & 0 & 0 & 0 & 0 & 0 & 1.6 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.9 & 0 & 0 & 0 & 0 & 0 & 8.0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.3 & 0 & 0 & 0 & 0 & 0 & 0 & 11. & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.1 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20
\end{array}\right].$$

For the following example, let $L = [2, 4, 6, 8, \ldots, 40]$, and $M = [1, 3, 5, 7, \ldots, 39]$.

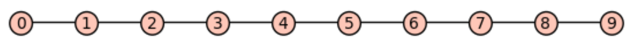
8. G is the following tree on 20 vertices, and $i = 2$.



$$A \approx \left[\begin{array}{cc|c|ccccccccccccccccc}
39 & 0 & 2.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 6.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.5 & 0 & 1.9 & 0 & 0 & 0 & 0 & 0 \\
\hline
2.2 & 0 & 21 & 0 & 0 & 0 & 0 & 0 & 0 & 9.8 & 0 & 0 & 0 & 9.4 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 6.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 30. & 0 & 0 & 0 & 0 & 0 & 0 & 1.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 29 & 0 & 0 & 0 & 5.0 & 0 & 4.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 11 & 0 & 0 & 0 & 0 & 0 & 1.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 33 & 0 & 0 & 0 & 1.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 25 & 0 & 1.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 9.8 & 0 & 0 & 5.0 & 0 & 0 & 0 & 28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.5 & 0 & 25 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1.9 & 4.6 & 0 & 1.8 & 0 & 0 & 0 & 29 & 0 & 0 & 0 & 0 & 0 & 3.1 & 1.6 & 0 \\
0 & 2.5 & 0 & 0 & 0 & 0 & 1.4 & 0 & 0 & 0 & 0 & 0 & 8.1 & 3.8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 9.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.8 & 12 & 0 & 1.5 & 2.5 & 0 & 0 & 0 \\
0 & 1.9 & 0 & 1.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6.2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.5 & 0 & 18 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.5 & 0 & 0 & 15 & 0 & 0 & 1.1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0 & 3.1 & 0 & 0 & 0 & 0 & 0 & 26 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.6 & 0 & 0 & 0 & 0 & 0 & 0 & 36 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.1 & 0 & 0 & 15
\end{array}\right].$$

A.2.2 A case study on paths on 10 vertices

In Section A.2.1 we saw an example of a path on 10 vertices where the deleted vertex was a non-pendent vertex. Below, we study how fast the algorithm terminates with respect to the vertex deleted. Let $L = [-10, -8, -5, -1, 2, 5, 7, 10, 12, 15]$, $M = [-9, -6, -3, 1, 3, 6, 9, 11, 13]$, and A_i represents the solution to the λ-μ-SIEP when the i-th vertex is deleted.



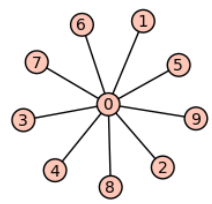
Consider the following table which compares the time to operate the algorithm. The 'time per loop's are chosen as the best of 3. We expect that it takes the same time for the algorithm to terminate when j or $9 - j$ are deleted.

Vertex deleted	Number of loops	Time per loop
0	5	5.200 s
1	5	1.940 s
2	5	0.784 s
3	5	0.374 s
4	5	0.234 s
5	5	0.236 s
6	5	0.372 s
7	5	0.788 s
8	5	1.940 s
9	5	5.170 s

The results show that for the pendent vertices it takes the longest, and for the vertices which are closer to the 'center' of the graph it takes shorter time for the algorithm to terminate. This suggests that the algorithm time depends on the number of times that the function is being recalled.

A.2.3 A case study on stars on 10 vertices

In Section A.2.1 we saw two examples of a star on 10 vertices where once the deleted vertex was the non-pendent vertex, and other time the deleted vertex was a pendent vertex. Below, we study how fast the algorithm terminates with respect to the vertex deleted. Let $L = [-10, -8, -5, -1, 2, 5, 7, 10, 12, 15]$, $M = [-9, -6, -3, 1, 3, 6, 9, 11, 13]$, and A_i represents the solution to the λ-μ-SIEP when the i-th vertex is deleted.



Consider the following table which compares the time to operate the algorithm. The 'time per loop's are chosen as the best of 3. We expect that it takes the same time for the algorithm to terminate when the deleted vertices are pendent vertices, and we expect it to be longer than when the deleted vertex is the center of the star.

Vertex deleted	Number of loops	Time per loop
0	25	0.0102 s
1	5	0.1380 s
2	5	0.1360 s
3	5	0.1370 s
4	5	0.1360 s
5	5	0.1370 s
6	5	0.1350 s
7	5	0.1350 s
8	5	0.1360 s
9	5	0.1360 s

A.3 The `lambda_tau_for_trees()` function

The following SAGE code defines a function `lambda_tau_for_trees()`, where G is a tree on n vertices, L is a list of n distinct real numbers, T is a list of $n-2$ distinct real numbers, and r and s are distinct integer between 0 and $n-1$. The output of the function is an $n \times n$ real symmetric matrix A such that $\mathcal{G}(A) = G$, $\sigma(A) = L$, and $\sigma(A(\{r,s\})) = T$.

```
###########################################################
# We have to redefine the lambda_mu_for_tees() function in
# order to be able to use it correctly to solve the
# lambda-tau problem. The problem with the old version of
# lambda_mu_for_trees() is that it doesn't take care of
# indices of vertices in a way that we need in the
# lambda_tau_for_trees(). So here is another version of
# lambda_mu_for_trees().
###########################################################
# This checks to see if the input is correct, and if it is
# not it prints a meaningful error message.
# This is a boolean function with outputs True and False
def is_input_for_lambda_mu(G,L,M,i):
# G is a tree on n vertices with vertex i
# L is a list of n distinct real numbers
# M is a list of n-1 distinct real numbers that strictly
# interlaces L
# this function checks to so if the above are actually true
    Error = 0
  # this simply indicates which error message to show
    Errors = ["Calculating...", # 0, nothing is wrong
             "There are duplicate lambda's!", # 1
             "There are duplicate mu's!", # 2
             "Some lambda is equal to some mu!", # 3
```

```
              "Number of lambda's should be exactly one more than the
              [+] number of mu's!", # 4
              "'i' should be a vertex of the graph , that is, an
              [+]integer between 0 and %s, but it is %s." %(len(L)-1,
              [+]i), #5
              "G should be either a tree or the adjacency matrix of a
              [+] tree!", # 6
              "Number of eigenvalues should be equal to the number of
              [+] the vertices", # 7
              "Interlacing inequalities are not met!", # 8
              ]
    if len(L) > len(set(L)):
  # Are lambda's distinct?
        Error = 1
    elif len(M) > len(set(M)):
  # Are mu's distinct?
        Error = 2
    elif len(L+M) > len(set(L+M)):
  # Are lambda's different from mu's?
        Error = 3
    elif len(L) != len(M) + 1:
  # Are lambda's one more than mu's?
        Error = 4
    elif not G.has_vertex(i):
  # Is i a vertex of G?
        Error = 5
    elif not G.is_tree():
        Error = 6
    elif len(L) != G.size()+1:
        Error = 7
    else:
```

```
  # Are the Cauchy interlacing inequalities met?
        L.sort()
        M.sort()
        for l in range(len(L)-1):
            if M[l] < L[l] or M[l] > L[l+1]:
                Error = 8
    if Error != 0:
  # Return the error message, if any.
        print Errors[Error]
        print ('---------------------------------------')
        print ('---------------No output!--------------')
        print ('---------------------------------------')
        return(False)
    else:
        return(True)
###########################################################
# This gets two lists, adds the elements of each list, and
# returns the difference of the two sums
def difference_of_sums(L,M):
# L is list of numbers
# M is list of numbers
    sumL = 0
    sumM = 0
    for l in range(len(L)): # Add up lambda's
        sumL = sumL + L[l]
    for m in range(len(M)): # Add up mu's
        sumM = sumM + M[m]
    return(sumL - sumM)
###########################################################
# This finds the connected components of G after deleting
# vertex i and returns I =[[H_j, e_j] for j ], where H_j are
```

```
# connected components of G(i) and e_j is the vertex of H_j
# which was a neighbors of i in G.
def list_of_connected_components(G,i):
# G is a tree with a vertex i
    # Make a copy of G and delete the i-th vertex of it and
  # call it H
    H = copy(G)
    H.delete_vertices([i])
    CCH = H.connected_components()
    N = G.neighbors(i) # Get a list of nbrs of vertex i in G
    I = []
  # a list of connected components of [G(i),v] where
  # v is the neighbor of i in G and in that component
    for cc in range(len(CCH)):
        e = (set(CCH[cc])).intersection(set(N)).pop()
    # this intersection has only one element and I want
    # to look at that one element.
        I = I + [[H.subgraph(CCH[cc]),e]]
    return(I)
###########################################################
# This looks lists all the numerators of the partial
# fraction decompositions of prod(x - l_i) / prod(x - m_i)
def numerators_of_pfd(L,M):
    B = []
  # f/g = (x-a) - sum b/(x - u) and B is a list of all b's
    for m in range(len(M)):
        top = 1
        bottom = 1
        for j in range(len(L)):
            top = top * (M[m] - L[j])
        for j in range(len(M)):
```

```
            if j != m:
                bottom = bottom * (M[m] - M[j])
        B = B + [- top / bottom]
    return(B)
##########################################################
# This gets two numbers and two lists and returns the square
# root of the coefficient of y_j along with some other
# things that are gonna be used in further calculations
def coefficient_of_yj(count,s,M,B):
    tempL = []
  # mu's will be broken into lists of size s and each list
  # will be assigned as new lamba's
    tempF = 0
  # This is a rational function that adds enough
  # b / (x - mu) for each component
    for l in range(count, count + s):
  # goes through this connected component
        tempL = tempL + [M[l]]
        tempF = tempF + B[l] / (x - M[l])
    v = ((tempF.numerator()).expand()).leading_coefficient(x)
    w = ((tempF.denominator()).expand()).leading_coefficient(x)
    return(sqrt(v/w),tempF,tempL)
##########################################################
# This piece solves the lambda-mu SIEP for trees
# This is a version of lambda_mu_for_tree() that looks at
# the indices of the graph and if the largest index is n
# (counted from zero), then it returns a (n+1)x(n+1) matrix
# and the indices that are missing in the graph will have
# zero rows and columns at the end. This is because later I
# want to add the missing vertices from another graph.
def lambda_mu_for_tree(G,L,M,i,Out=None):
```

```
# G is a graph or the adjacency matrix of a graph. In case
# that it is a matrix, it's type should match
# 'sage.matrix.matrix_integer_dense.Matrix_integer_dense'
# L is a litst of distinct
# real numbers to be realized
# as eigenvalues of A
# M is a list of distinct
# real numbers to be realiezed
# as eigenvalues of A(i)
# i is the vertex to be deleted
    if type(G) == sage.matrix.matrix_integer_dense.
    [+]Matrix_integer_dense:
        G = Graph(G)
    indices = list(G)
    order = max(indices)+1
    if Out is None:
        Out = [[0 for j in range(order)] for j in range(order)]
    if is_input_for_lambda_mu(G,L,M,i): # check the inputs
  # Here I want to evaluate the diagonal entry (i,i)
        if len(L) == 1:
    # If there is only one lambda, then that's the
    # diagonal entry
            a = copy(L[0])
            Out[i][i] = copy(a)
            Out = Matrix(RR,Out)
      # I'm not deleting extra rows and columns since
      # I'll need them in lambda_tau problem
            return(Out)
        # If there are more than one lambda, then the
    # diagonal entry is the difference of traces (sum of
    # lambda's minus sum of mu's)
```

```
        a = difference_of_sums(L,M)
        Out[i][i] = copy(a)
        I = list_of_connected_components(G,i)
        B = numerators_of_pfd(L,M)
        count = 0
        for component in range(len(I)):
    # remember 'I' was a list of connected components
            s = I[component][0].size() + 1
      # +1 because size() starts from 0
            e = I[component][1]
            # this gets s, M, B and returns the sqrt(v/w)
            (a,tempF,tempL) = coefficient_of_yj(count,s,M,B)
            # Write the i,j_i entry of the output matrix
            Out[i][e] = Out[e][i] = a
      # this is A_{i, j_i} entry
            # Here I find the roots of the numerator of the
      # sums of b / (x - mu)
            if (tempF.numerator()).degree(x) == 0:
                tempM = []
            else:
                # we know all the solution are real, but
        # there are calculation errors, so we'll
        # find all the complex solutions and then
        # omit the very small imaginary parts
                S = (tempF.numerator()).roots(ring = CC,
                [+]multiplicities = False)
                for solution in range(len(S)):
                    S[solution] = S[solution].real()
                tempM = copy(S)
            count = count + s
            # Here I call lambda_mu_for_tree() for subtree
```

```
   # 'component'
      lambda_mu_for_tree(I[component][0], tempL, tempM, e, Out[+])
    Out = Matrix(RR,Out)
  # I'm not deleting extra rows and columns since I'll
  # need them in lambda_tau problem
    return(Out)
#############################################################
# This shows the output matrix, its eigenvalues and the
# eigenvlaues of A(i), and its graph
def check_output_lambda_mu_for_tree(A,i,precision=8):
# A is a matrix which is the output of lambda_mu_for_trees()
# i is the one that also is entered in lambda_mu_for_trees()
# precision is an integer that shows how many digits do I
# want to be printed at the end. I set the default to be 8
  eigA = A.eigenvalues()
  EigA = []
  for e in eigA:
    EigA = EigA + [e.n(precision)]
  eigB = (A.delete_rows([i]).delete_columns([i])).eigenvalues()
  EigB = []
  for e in eigB:
    EigB = EigB + [e.n(precision)]
  print('A is:')
  print(A.n(precision))
  print(' ')
  print('Eigenvalues of A are: %s') %(EigA)
  print(' ')
  print('Eigenvalues of A(%s) are: %s') %(i,EigB)
  AdjA = matrix(A.ncols())
  for i in range(A.ncols()):
```

```
        for j in range(A.ncols()):
            if i != j:
                if A[i,j] != 0:
                    AdjA[i,j] = 1
    FinalGraph = Graph(AdjA)
    print(' ' )
    print('And the graph of A is:')
    FinalGraph.show()
###########################################################
# Here is a sample input
G = Graph(graphs.PetersenGraph().min_spanning_tree())
L = [-3,-1,2,4,6,8,11,15,19,23]
M = [-2,1,3,5,7,10,12,17,21]
i = 1
A=lambda_mu_for_tree(G,L,M,i)
check_output_lambda_mu_for_tree(A,i,precision=12)
###########################################################
# This checks to see if the input is correct, and if it is
# not it prints a meaningful error message.
# This is a boolean function with outputs True and False
def is_input_for_lambda_tau(G,L,T,r,s):
# G is a tree or the adjacency matrix of a tree.
# L is list of distinct real numbers, the eigenvalues of A
# T is list of distinct real numbers, the eigenvalues of A({r,s})
# r and s are the vertices of G to be deleted
    Error = 0
  # this simply indicates which error message to show
    Errors = ["Calculating...", # 0, nothing is wrong
             "There are duplicate lambda's!", # 1
             "There are duplicate tau's!", # 2
             "Some lambda is equal to some tau!", # 3
```

```
             "Number of lambda's should be exactly two more than the
             [+] number of tau's!", # 4
             "'{r,s}' should be and edge of the graph!", #5
             "G should be either a tree or the adjacency matrix of a
             [+] tree!", # 6
             "Number of eigenvalues should be equal to the number of
             [+] the vertices", # 7
             "Second order interlacing inequalities are not met!", #
             [+] 8
             ]
    if len(L) > len(set(L)):
  # Are lambda's distinct?
        Error = 1
    elif len(T) > len(set(T)):
  # Are mu's distinct?
        Error = 2
    elif len(L+T) > len(Set(L+T)):
  # Are lambda's different from tau's?
        Error = 3
    elif len(L) != len(T) + 2:
  # Are lambda's two more than tau's?
        Error = 4
    elif not G.has_edge(r,s):
  # Check that {r,s} is an edge of the graph (otherwise
  # deleting it does not produce any errors)
        Error = 5
    elif not G.is_tree():
        Error = 6
    elif len(L) != G.size()+1:
        Error = 7
    else:
```

```
  # Are the Cauchy interlacing inequalities met?
        L.sort()
        T.sort()
        for l in range(len(L)-2):
            if T[l] < L[l] or T[l] > L[l+2]:
                Error = 8
    if Error != 0:
  # Return the error message, if any.
        print Errors[Error]
        print ('--------------------------------------')
        print ('---------------No output!-------------')
        print ('--------------------------------------')
        return(False)
    else:
        return(True)
##########################################################
# This partitions two lists into four lists
def partition_mu_tau(M,T,n1,n2):
    TauPairings = []
    MuPairings = []
    M.sort()
    T.sort()
    MT = M + T
    MT.sort()
    newMT = copy(MT)
    M1 = []
    M2 = []
    T1 = []
    T2 = []
    k=0
    for i in range(0,len(MT)-1):
```

```
# I can make bounds even smaller since first
# and last things can't be a tau pairing
      a = copy(MT[i])
      b = copy(MT[i+1])
      if a in T and b in T:
          k=k+1
          TauPairings = TauPairings + [a,b]
          T.remove(a)
          T.remove(b)
          newMT.remove(a)
          newMT.remove(b)
          T1 = T1 + [a]
          T2 = T2 + [b]
          i = i + 1
    #Because if there is a tau pairing, the next one
    # is not a tau or...
      if a in M and b in M:
          MuPairings = MuPairings + [a,b]
          i = i + 1
    #Because if there is a lambda pairing, the next
    # one is not a lambda or...
          M.remove(a)
          M.remove(b)
          newMT.remove(a)
          newMT.remove(b)
          M1 = M1 + [a]
          M2 = M2 + [b]
  while len(M1) < n1:
      a = newMT[0]
      b = newMT[1]
      if a in M:
```

```
            M1 = M1 + [a]
            T1 = T1 + [b]
        else:
            M1 = M1 + [b]
            T1 = T1 + [a]
        newMT.remove(a)
        newMT.remove(b)
    while len(M2) < n2:
        a = newMT[0]
        b = newMT[1]
        if a in M:
            M2 = M2 + [a]
            T2 = T2 + [b]
        else:
            M2 = M2 + [b]
            T2 = T2 + [a]
        newMT.remove(a)
        newMT.remove(b)
    return(M1,T1,M2,T2)
###########################################################
# ebsilon is the maximum of negative local extrema of f/g.
# This finds ebsilon where L,T are roots of f and g.
def maximum_negative_local_extrema(L,T):
    var('x')
    f = 1
    for i in range(len(L)):
        f = f * (x - L[i])
    g = 1
    for i in range(len(T)):
        g = g * (x- T[i])
    h = f/g
```

```
    hprime = h.diff(x)
    hzegond = hprime.diff(x)
    Hprime = (hprime.numerator()).roots(ring = RR, multiplicities =
    [+]False)
    LocalMaxima = []
    for i in range(len(Hprime)):
        t = h(Hprime[i])
        if t < 0:
            if abs(hzegond(Hprime[i])) > 0:
      # Here I'm a little concerned that it might be
      # very close to zero but not exactly zero due to
      # precision errors. So if that occures I need to
      # think of a good way to increase 0 to something
      # larger. That's why I'm using abs() > 0,
      # instead of != 0.
                LocalMaxima = LocalMaxima + [t]
    return(h,max(LocalMaxima))
##################################################
# This piece solves the lambda-tau SIEP for trees using the
# lambda_mu_for_tree() function above and the
# partition_mu_tau() function above.
def lambda_tau_for_tree(G,L,T,r,s):
# G is a tree or the adjacency matrix of a tree. If it is a
# matrix, its type should match
# 'sage.matrix.matrix_integer_dense.Matrix_integer_dense'
# L is list of distinct real numbers, the eigenvalues of A
# T is list of distinct real numbers, the eigenvalues of A({r,s})
# r and s are the vertices of G to be deleted
    if type(G) == sage.matrix.matrix_integer_dense.
    [+]Matrix_integer_dense:
        G = Graph(G)
```

```
    if is_input_for_lambda_tau(G,L,T,r,s):
  # check if input is correct
  # delete the edge {r,s} from the tree and find the two
  # connected components G1 (containing r) and G2
  # (containing s)
        H = copy(G)
        H.delete_edge(r,s)
  # After deleting the edge (r,s) there are two connected
  # components, I'll make a list of them and figure out
  # which one has r in it, and which one has s in it
        I=[]
    # I'm making a list of connected components of
    # G({r,s}), well, there are two of them
        for component in H.connected_components():
                I = I + [H.subgraph(component)]
        if I[0].has_vertex(r):
            G1 = I[0]
            G2 = I[1]
        else:
            G1 = I[1]
            G2 = I[0]
# Find number of vertices in G1 and G2
        n1 = G1.size() + 1 # It counts things from 0
        n2 = G2.size() + 1 # It counts things from 0
# Now count the number of tau pairings, k, and check if
# n1,n2 > k # I could have done this inside the function
# that partitions the M and T, but I don't want to call it
# if there is no solution.
        k = 0 # Number of tau pairings
        LT = L+T
        LT.sort()
```

```
        for i in range(1,len(LT)-1):
    # I can make bounds even smaller since first and
    # last things can't be a tau pairing
            if LT[i] in T and LT[i+1] in T:
                k=k+1
                i = i + 1
        #Because if there is a tau pairing, the next
        # one is not a tau
        if n1 < k or n2 < k:
            print("There is no solution for this problem, since the
            [+]number of the vertices in each side of 'r' and 's'
            [+]need to be greater than the number of tau-pairings: %
            [+]s. But the number of vertices on the two sides are %s
            [+] and %s.") %(k,n1,n2)
            print ('---------------------------------------')
            print ('--------------No Solution!-------------')
            print ('---------------------------------------')
            return()
  # a_{r,s} can varry as much as ebsilon, where ebsilon is
  # the maximum of negative local extrema of f/g. This
  # finds ebsilon where L,T are roots of f and g.
        (h,m) = maximum_negative_local_extrema(L,T)
        ebsilon = min(-9/10 * m, (1)^2)
    # I'm choosing this because sometimes |m| is
    # big and I don't want a huge entry in my matrix
        # find M: the roots of h + ebsilpn
        M = ((h + ebsilon).numerator()).roots(ring = RR,
        [+]multiplicities = False)
        # call partition_mu_tau() in order to get two lists
        (M1,T1,M2,T2) = partition_mu_tau(M,T,n1,n2)
        # now call lambda_mu_for_tree(G1,M1,T1,r)
```

```
        A1 = lambda_mu_for_tree(G1,M1,T1,r)
        A2 = lambda_mu_for_tree(G2,M2,T2,s)
    # A1 and A2 don't have correct sizes, I just add
    # enough zeros to make them as big as desired
        B1 = matrix(RR,len(L))
        B2 = matrix(RR,len(L))
        for i in range(A1.ncols()):
            for j in range(A1.ncols()):
                B1[i,j] = A1[i,j]
        for i in range(A2.ncols()):
            for j in range(A2.ncols()):
                B2[i,j] = A2[i,j]
    # now it's time to put all of them together and
    # ebsilon on the (r,s) entry to get the final matrix
        Out = B1 + B2
        Out[r,s] = Out[s,r] = sqrt(ebsilon)
        return(Out)
####################################################
# This shows the output matrix, its eigenvalues and the
# eigenvlaues of A(i), and its graph
def check_output_lambda_tau_for_tree(A,r,s,precision=8):
# A is a matrix which is the output of lambda_mu_for_trees()
# i is the one that also is entered in lambda_mu_for_trees()
# precision is an integer that shows how many digits do I
# want to be printed at the end. I set the default to be 8
    eigA = A.eigenvalues()
    EigA = []
    for e in eigA:
        EigA = EigA + [e.n(precision)]
    eigB = (A.delete_rows([r,s]).delete_columns([r,s])).eigenvalues
    [+]()
```

```
    EigB = []
    for e in eigB:
        EigB = EigB + [e.n(precision)]
    print('A is:')
    print(A.n(precision))
    print(' ')
    print('Eigenvalues of A are: %s') %(EigA)
    print(' ')
    print('Eigenvalues of A({%s,%s}) are: %s') %(r,s,EigB)
    AdjA = matrix(A.ncols())
    for i in range(A.ncols()):
        for j in range(A.ncols()):
            if i != j:
                if A[i,j] != 0:
                    AdjA[i,j] = 1
    FinalGraph = Graph(AdjA)
    print(' ' )
    print('And the graph of A is:')
    FinalGraph.show()
```

Code A.3: The `lambda_tau_for_trees()` function.

References

[1] B. A. Bolt, "What can inverse theory do for applied mathematics and the sciences?" *Australian Mathematical Society Gazette*, vol. 7, pp. 69–78, 1980.

[2] M. T. Chu and G. H. Golub, "Structured inverse eigenvalue problems," *Acta Numerica*, pp. 1–71, 2002.

[3] A. L. Duarte, "Construction of acyclic matrices from spectral data," *Linear Algebra and its Applications*, vol. 113, pp. 173–182, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0024379589902954

[4] M. Spivak, *Calculus On Manifolds: A Modern Approach To Classical Theorems Of Advanced Calculus*. Westview Press, 1971.

[5] S.-G. Hwang, "Cauchy's interlace theorem for eigenvalues of hermitian matrices," *The American Mathematical Monthly*, vol. 111, no. 2, pp. pp. 157–159, 2004. [Online]. Available: http://www.jstor.org/stable/4145217

[6] D. Kalman, "A matrix proof of Newton's identities," *Mathematics Magazine*, vol. 73, pp. 313–315, 2000.

[7] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed. Baltimore, MD: Johns Hopkins University Press, 1996.

[8] B. N. Parlett, *The Symmetric Eigenvalue Problem, Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics, 1987.

[9] M. T. Chu and G. H. Golub, *Inverse Eigenvalue Problems: Theory, Algorithms, and Applications*, ser. Numerical Mathematics and Scientific Computation. Oxford University Press, 2005.

[10] G. M. L. Gladwell, *Inverse Problems in Vibration*. Dordrecht, Netherlands: Martinus Nijhoff, 1986.

[11] H. Hochstadt, "Asymptotic estimates of the Sturm-Liouville spectrum," *Comm. Pure Appl. Math.*, vol. 14, pp. 749–764, 1961.

[12] ——, "On some inverse problems in matrix theory," *Arch. Math.*, vol. 18, pp. 201–207, 1967.

[13] ——, "The inverse Sturm-Liouville problem," *Comm. Pure Appl. Math.*, vol. 26, pp. 715–729, 1973.

[14] ——, "On the construction of a Jacobi matrix from spectral data," *Linear Algebra Appl..*, vol. 8, pp. 435–446, 1974.

[15] ——, "On the construction of a Jacobi matrix from mixed given data," *Linear Algebra Appl..*, vol. 28, pp. 113–115, 1979.

[16] O. H. Hald, "On discrete and numerical inverse Sturm-Liouville problems," Ph.D. dissertation, New York University, 1972.

[17] ——, "Inverse eigenvalue problems for Jacobi matrices," *Linear Algebra and Appl.*, vol. 14, no. 1, pp. 63–85, 1976.

[18] L. Gray and D. Wilson, "Construction of a Jacobi matrix from spectral data," *Linear Algebra Appl.*, vol. 14, pp. 131–134, 1976.

[19] C. de Boor and G. H. Golub, "The numerically stable reconstruction of a Jacobi matrix from spectral data," *Linear Algebra Appl.*, vol. 21, pp. 245–260, 1978.

[20] G. N. de Oliveira, "Matrices with prescribed characteristic polynomial and a prescribed submatrix," *Pacific J. Math.*, vol. 29, pp. 14–21, 1969.

[21] K. Fan and G. Pall, "Imbedding conditions for Hermitian and normal matrices," *Canad. J. Math.*, vol. 9, pp. 298–304, 1957.

[22] F. R. Gantmacher and M. G. Kreĭn, *Oszillationsmatrizen, Oszillationskerne und kleine schwingungen mechanischer Systeme (Oscillation matrices and kernels and small oscillations of mechanical systems).* Berlin, Akademie-Verlag, 1960, 1960, (in German).

[23] L. J. Gray and D. G. Wilson, "Construction of a Jacobi matrix from spectral data," *Linear Alg. Appl.*, vol. 14, pp. 131–134, 1976.

[24] K. Löwner, "Über monotone matrixfunktion," *Math Z.*, vol. 38, pp. 177–216, 1934.

[25] L. Mirsky, "Matrices with prescribed characteristic roots and diagonal elements," *J. London Math. Soc.*, vol. 33, pp. 14–21, 1958.

[26] G. E. Shilov, *An introduction to linear spaces.* Englewood Cliffs, NJ: Prentice-Hall, 1961, (transl. of the Russian ed., Moscow, 1952).

[27] T. J. Stieltjes, "Recherches sur les fractions continues," *Ann. Fac. Sci. Toulouse*, vol. 8J, pp. 1–122, 1894.

[28] ——, "Recherches sur les fractions continues," *Ann. Fac. Sci. Toulouse*, vol. 9A, pp. 1–47, 1895.

[29] R. C. Thompson, "Principal submatrices of normal and Hermitian matrices," *Illinois J. Math.*, vol. 10, pp. 296–308, 1966.

[30] ——, "The behavior of eigenvalues and singular values under perturbations of restricted rank," *Linear Algebra Appl.*, vol. 13, pp. 69–78, 1976.

[31] B. N. Parlett, *The symmetric eigenvalue problem.* Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 1998, corrected reprint of the 1980 original.

[32] P. Nylen, "Inverse eigenvalue problem: existence of special mass-damper-spring systems," *Linear Algebra Appl.*, vol. 297, no. 1-3, pp. 107–132, 1999.

[33] Y. H. Zhang, "A note on the sufficient conditions for the solvability of general algebraic inverse eigenvalue problems," *Shandong Daxue Xuebao Ziran Kexue Ban*, vol. 34, no. 2, pp. 139–143, 1999.

[34] L. Zhang, D. X. Xie, and X. Y. Hu, "The inverse eigenvalue problems of bisymmetric matrices on the linear manifolds," *Math. Numer. Sin.*, vol. 22, no. 2, pp. 129–138, 2000.

[35] B. N. Datta, S. Elhay, Y. M. Ram, and D. R. Sarkissian, "Partial eigenstructure assignment for the quadratic pencil," *J. Sound Vibration*, vol. 230, no. 1, pp. 101–110, 2000.

[36] M. T. Chu, "A fast recursive algorithm for constructing matrices with prescribed eigenvalues and singular values," *SIAM J. Numer. Anal.*, vol. 37, no. 3, pp. 1004–1020 (electronic), 2000.

[37] M. Arav, D. Hershkowitz, V. Mehrmann, and H. Schneider, "The recursive inverse eigenvalue problem," *SIAM J. Matrix Anal. Appl.*, vol. 22, no. 2, pp. 392–412 (electronic), 2000.

[38] L. Starek and D. J. Inman, "Symmetric inverse eigenvalue vibration problem and its applications," *Mechanical Systems and Signal Processing*, vol. 15, no. 1, pp. 11–29, 2001.

[39] J. E. Motershead and M. I. Friswell, "Special issue of mechanical system and signal processing," in *Inverse Problem on Structural Dynamics (Liverpool, 1999).* Berlin: Academic Press, 2001, pp. 1–1.

[40] C. K. Li and R. Mathias, "Construction of matrices with prescribed singular values and eigenvalues," *BIT*, vol. 41, no. 1, pp. 115–126, 2001.

[41] E. Foltete, G. M. L. Gladwell, and G. Lallement, "On the reconstruction of a damped vibrating system from two complex spectra, part ii - experiment," *J. Sound Vib.*, vol. 240, pp. 219–240, 2001.

[42] C. R. Johnson, A. L. Duarte, and C. M. Saiago, "Inverse eigenvalue problems and lists of multiplicities of eigenvalues for matrices whose graph is a tree: the case of generalized stars and double generalized stars," *Linear Algebra*

and its Applications, vol. 373, pp. 311–330, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379503005822

[43] C. Johnson and C. S. A. Leal Duarte, "The parter-wiener theorem: refinement and generalization," *SIAM Journal of Matrix Analysis and Applications*, 2003.

[44] F. Barioli and S. M. Fallat, "Oi two conjectures regarding an inverse eigenvalue problem for acyclic symmetric matrices," *Electronic Journal of Linear Algebra*, vol. 11, pp. 41–50, 2004.

[45] I.-J. Kim and B. L. Shader, "Unordered multiplicity lists of a class of binary trees," *Linear Algebra and its Applications*, vol. 438, pp. 3781–3788, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379511005180

[46] K. H. Monfared and B. L. Shader, "Construction of matrices with a given graph and prescribed interlaced spectral data," *Linear Algebra and its Applications*, vol. 438, no. 11, pp. 4348–4358, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379513001006

[47] S. G. Krantz and H. R. Parks, *The Implicit Function Theorem: History, Theory, and Applications*, ser. Modern Birkhäuser Classics. Springer, 2013. [Online]. Available: http://www.springer.com/birkhauser/mathematics/book/978-1-4614-5980-4

[48] W. Barrett, A. Lazenby, N. Malloy, C. Nelson, W. Sexton, R. Smith, J. Sinkovic, and T. Yang, "The combinatorial inverse eigenvalue problem: Complete graphs and small graphs with strict inequality," *Electronic Journal of Linear Algebra,*, vol. 26, pp. 656–672, 2013.

[49] W. Barrett, C. Nelson, J. Sinkovic, and T. Yang, "The combinatorial inverse eigenvalue problem ii: All cases for small graphs," *(submitted)*.

[50] R. Horn and C. Johnson, *Matrix Analysis*. New York: Cambridge University Press, 1991.

[51] H. van der Holst, L. Lovász, and A. Schrijver, "The Colin de Verdiére graph parameter," *Graph Theory and Combinatorial Biology*, vol. 7, pp. 29–85 *Bolyai Soc. Math. Stud, Budapest*, 1999.

[52] Y. C. de Verdiére, "Sur un nouvel invariant des graphes et un critère de planarité," *Journal of Combinatorial Theory, Series B*, vol. 50, pp. 11–21 [English translation: On a new graph invariant and a criterion for planarity in: *Graph Structure Theory* (N. Robertson, P. Seymour, eds.), Contemporary Mathematics, American Mathematical Society, Providence, Rhode Island, 1993, pp. 137–147], 1990.

[53] V. Guillemin and A. Pollack, *Differential Topology*, ser. AMS Chelsea Publishing. American Mathematical Society, 2010.

[54] R. Horn and C. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, 1991.

[55] A. E. Brouwer and W. H. Haemers, *Spectra of graphs*. Springer, 2011.

[56] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge University Press, 1991.

[57] I.-J. Kim and B. L. Shader, "On Fiedler- and Parter-vertices of acyclic matrices," *Linear Algebra and its Applications*, vol. 428, pp. 2601–2613, 2008.

[58] D. Boley and G. H. Golub, "A survey of matrix inverse eigenvalue problems," *Inverse Problems*, vol. 3, no. 4, pp. 595–622, 1987.

[59] M. T. Chu, "Inverse eigenvalue problems," *SIAM Rev.*, vol. 40, no. 1, pp. 1–39 (electronic), 1998.

[60] C. R. Johnson, J. Nuckols, and C. Spicer, "The implicit construction of multiplicity lists for classes of trees and verification of some conjectures," *Linear Algebra and its Applications*, vol. 438, pp. 1990–2003, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379512008014

[61] C. Johnson and A. Leal-Duarte, "The maximum multiplicity of an eigenvalue in a matrix whose graph is a tree," *Linear and Multilinear Algebra*, vol. 46, pp. 139–144, 1999.

[62] ——, "On the possible multiplicities of the eigenvalues of an hermitian matrix whose graph is a given tree," *Linear Algebra Appl.*, vol. 348, pp. 7–21, 2002.

[63] C. Johnson, A. Leal-Duarte, C. Saiago, B. Sutton, and A. Witt, "On the relative position of multiple eigenvalues in the spectrum of an hermitian matrix with a given grap," *Linear Algebra Appl.*, vol. 363, pp. 147–159, 2003.

[64] C. Johnson and C. Saiago, "Estimation of the maximum multiplicity of an eigenvalue in terms of the vertex degrees of the graph of a matri," *Electron. J. Linear Algebra*, vol. 9, pp. 27–31, 2002.

[65] C. R. Johnson, B. D. Sutton, and A. J. Witt, "Implicit construction of multiple eigenvalues for trees," *Linear and Multilinear Algebra*, vol. 57, pp. 409–420, 2009. [Online]. Available: http://www.tandfonline.com/doi/pdf/10.1080/03081080701852756

[66] P. Oblaka and H. Śmigoc, "Graphs that allow all the eigenvalue multiplicities to be even," *Linear Algebra and its Applications*, vol. 454, pp. 72–90, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379514002316

[67] K. H. Monfared and S. Mallik, "Construction of real skew-symmetric matrices from interlaced spectral data, and graph," *(submitted)*.

[68] S. M. Fallat and C. R. Johnson, *Totally Nonnegative Matrices*, ser. Princeton Series in Applied Mathematics. Princeton University Press, 2011. [Online]. Available: http://press.princeton.edu/titles/9492.html

[69] C. Garnett and B. L. Shader, "The nilpotent-centralizer method for spectrally arbitrary patterns," *Linear Algebra and its Applications*, 2011, to appear, http://dx.doi.org/10.1016/j.laa.2011.10.004,.

[70] B. L. Shader and C. L. Shader, "A sign pattern that allows oppositely signed orthogonal matrices," *arXiv:*1212.6062 *[math.CO]*. [Online]. Available: http://arxiv.org/abs/1212.6062

[71] B. L. Shader, personal communication.