EULER MATRICES OF QUIVERS AND MAHLER MEASURE

TY NICHOLS

ABSTRACT. For a given quiver Q, we can define the matrix B such that $B = -(E^T)E^{-1}$, where E is the Euler matrix of Q. We will investigate the Mahler measure of the characteristic polynomial of B for small quivers, paying particular attention to quivers whose underlying graph is close to being a simply laced Dynkin diagram or an extended Dynkin diagram. We hope to use properties of B to investigate Lehmer's Conjecture. For this project, we will be working with Professor Harm Derksen.

1. INTRODUCTION

We begin by introducing basic terminology that motivates our approach to the problem. The main objects under study are quivers, and the related matrices are defined for use in the theory of quiver representations.

1.1. Quivers.

Definition 1. [1] A *quiver* Q is a 4-tuple $Q = (Q_0, Q_1, h, t)$ where Q_0 is a finite set of vertices, Q_1 is the set of edges, and $h : Q_1 \to Q_0$ and $t : Q_1 \to Q_0$ are functions that give the head and tail of each edge, respectively.

Definition 2. [1] A *path* on a quiver $Q = (Q_0, Q_1, h, t)$ from u to v for $u, v \in Q_0$ is some sequence of edges $e_0, \ldots e_n$ such that for $0 \le i \le n$, $h(e_{i-1}) = t(e_i)$ and $t(e_0) = u$, $h(e_n) = v$.

Example 1.1. Consider the quiver illustrated by the following diagram:
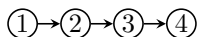
$$①\to②\to③\to④$$

Then $Q_0 = \{1, 2, 3, 4\}$, $Q_1 = \{(1, 2), (2, 3), (3, 4)\}$, and $h((1, 2)) = 2$ while $t((1, 2)) = 1$.

Example 1.2. Consider

Then $Q_0 = \{1, 2\}$, $Q_1 = \{(1, 2), (2, 1), (1, 1)\}$, and $h((1, 1)) = t((1, 1)) = 1$.

Notice that self-loops and cycles are permitted, and that the functions h and t are not necessarily injective, i.e. edges with the same head and tail may be repeated. However, we will be considering quivers that do not contain *oriented cycles*:

Definition 3. [1] An *oriented cycle* of a quiver $Q = (Q_0, Q_1, h, t)$ is a path from u to u. Notice that we may equivalently say that the cycle is a path from v to v for any v contained in the cycle.

Date: October 22, 2020.

Example 1.3. Consider again the quivers in Examples 0.1 and 0.2. Observe that the quiver in example 0.1 does not contain an oriented cycle, as there is no path from any node back to itself. However, the quiver in example 2 contains multiple oriented cycles; one consists only of the self-loop on 1, while the other consists of the edges $(1, 2)$ and $(2, 1)$.

1.2. **Quiver Representations.** The matrices under study for this project are defined in the study of quiver representations, which can be formally described as the following:

Definition 4. [1] A *representation* V of a quiver Q is a map V_0 that assigns a finite dimensional vector space to every vertex in Q_0, and for every edge $a \in Q_1$, an assignment of a linear transformation $V_a : V_0(t(a)) \to V_0(h(a))$.

Of interest in representation theory is the Euler Matrix of a quiver:

Definition 5. [1] Given a quiver $Q = (Q_0, Q_1, h, t)$, then the *Euler matrix* E of Q is defined as

$$E_{i,j} = \delta_{i,j} - |\{a \in Q_1 : t(a) = i, h(a) = j\}|$$
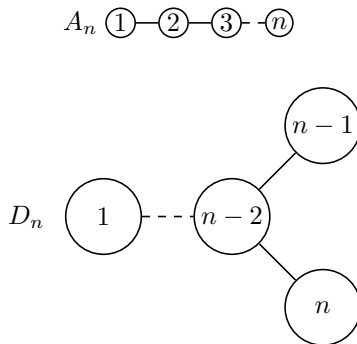
where $\delta_{i,j}$ is the Kronecker symbol, defined as

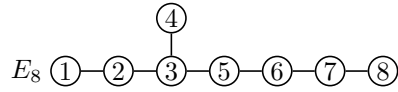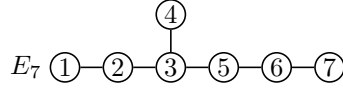$$\delta_{i,j} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}$$
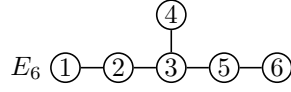
1.3. **Ausland-Reiter Transform.** This project studies the matrix product defined by $B = -E^T E^{-1}$, where E is the Euler Matrix for some quiver Q. The matrix B is used in the Ausland-Reiter Transform [1]; however, defining the transform is currently beyond the scope of this project.

1.4. **Dynkin Diagrams.** Of particular importance in representation theory are the *Dynkin diagrams*, which are related to irreducible root systems and Lie algebras [1]. Many of the connections in which these diagrams appear are beyond the scope of this paper, but the diagrams themselves will be important, so we provide them below.

Definition 6. [1]

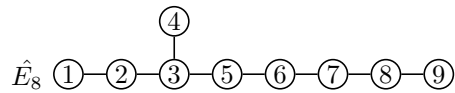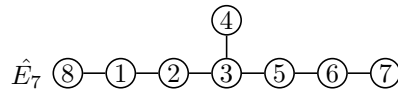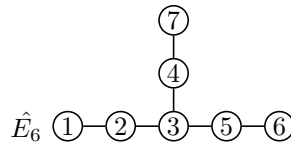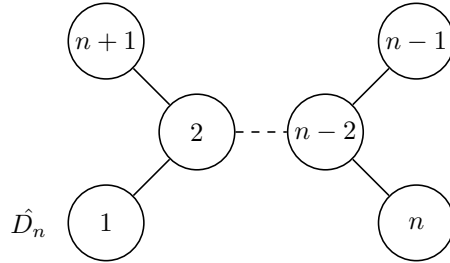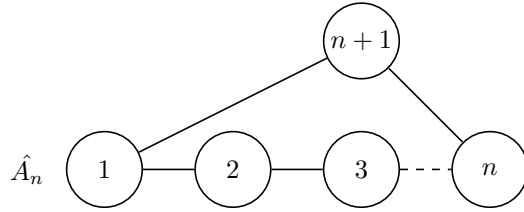The *simply laced Dynkin diagrams* are $A_n, D_n, E_6, E_7,$ and E_8 [1]. Dynkin diagrams are typically drawn as undirected graphs; if we speak of a quiver as being related to a Dynkin diagram, we mean the quiver drawn also as an undirected graph.

E_6

$$\begin{array}{c} \text{④} \\ \text{①}-\text{②}-\text{③}-\text{⑤}-\text{⑥} \end{array}$$

E_7

$$\begin{array}{c} \text{④} \\ \text{①}-\text{②}-\text{③}-\text{⑤}-\text{⑥}-\text{⑦} \end{array}$$

E_8

$$\begin{array}{c} \text{④} \\ \text{①}-\text{②}-\text{③}-\text{⑤}-\text{⑥}-\text{⑦}-\text{⑧} \end{array}$$

Also of interest are the *extended Dynkin diagrams* $\hat{A}_n$, $\hat{D}_n$, $\hat{E}_6$, $\hat{E}_7$, and $\hat{E}_8$.

$\hat{A}_n$

$\hat{D}_n$

$\hat{E}_6$

$$\begin{array}{c} \text{⑦} \\ \text{④} \\ \text{①}-\text{②}-\text{③}-\text{⑤}-\text{⑥} \end{array}$$

$\hat{E}_7$

$$\begin{array}{c} \text{④} \\ \text{⑧}-\text{①}-\text{②}-\text{③}-\text{⑤}-\text{⑥}-\text{⑦} \end{array}$$

$\hat{E}_8$

$$\begin{array}{c} \text{④} \\ \text{①}-\text{②}-\text{③}-\text{⑤}-\text{⑥}-\text{⑦}-\text{⑧}-\text{⑨} \end{array}$$

According to professor Derksen, the eigenvalues of B will lie on the unit circle when the quiver Q is either a simply laced Dynkin diagram or an extended Dynkin diagram.

Mahler Measure. We consider the *Mahler Measure* of polynomials to evaluate "closeness" of their roots to the unit circle in the complex plane:

Definition 7. [2]

Given a polynomial $P(x) \in \mathbb{Z}[x]$ with integer coefficients where $P(x) = a_n \Pi_{k=1}^n (x - \alpha_k)$, the *Mahler Measure of P* is defined as

$$M(P) = |a_d| \Pi_{k=1}^n \max\{1, |\alpha_k|\}$$

We will also define for convenience the Mahler Measure of a matrix, which will be the Mahler measure of its characteristic polynomial, and the Mahler measure of a quiver, which will be the Mahler measure of its matrix B when E is not singular.

For our matrix B, we may simplify the equation for Mahler measure slightly:

Remark 1.4. Since the matrix B has integer coefficients, its characteristic polynomial has integer coefficients, and since the characteristic polynomial has the eigenvalues of B as roots, the leading factor $|a_d|$ will always be 1. This means that

$$M(B) = \Pi_{k=1}^n \max\{1, |\lambda_k|\}$$

Observe that the Mahler Measure of a matrix is very similar to the determinant of that matrix, since $\det(B)$ is equal to the product of all eigenvalues of B. Knowing this, we can compare them:

Proposition 1.5. *For any matrix B with integer entries, $\det(B) \leq M(B)$.*

Proof. Suppose an nxn matrix B has eigenvalues $\lambda_1, \ldots \lambda_n$. Then $\det(B) = \lambda_1 \cdots \lambda_n$, so $|\det(b)| = |\lambda_1 \cdots \lambda_n| = |\lambda_1| \ldots |\lambda_n|$. But since $M(B) = |\max\{1, \lambda_1\}| \ldots |\max\{1, \lambda_n\}|$ and for any λ_i, $\lambda_i \leq \max\{1, \lambda_i\}$, it follows that $|\det(B)| \leq |M(B)|$.

However, Mahler measure is defined to be a positive real number, and B has integer entries, so $\det(B) \in \mathbb{Z}$. Therefore we can also say that $\det(B) \leq M(B)$. $\square$

Example 1.6. Consider the quiver A_4 given by

$$\textcircled{1} \rightarrow \textcircled{2} \rightarrow \textcircled{3} \rightarrow \textcircled{4}$$
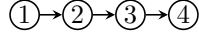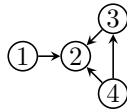
Then

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

and calculating we find that

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

We then calculate the eigenvalues for B, which are shown in the appendix; we find that $M(A_4) := M(B) = 1$.

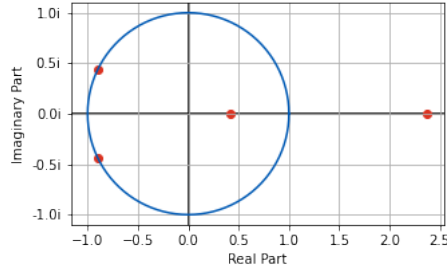**Example 1.7.** Consider the quiver A_4 given by

Then

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

and calculating we find that

$$B = \begin{pmatrix} -1 & -1 & 0 & 0 \\ 1 & 3 & 2 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & -2 & -1 & -1 \end{pmatrix}$$

We then calculate the eigenvalues for B, which are plotted on the complex plane below:

We can see that two of the eigenvalues are not on the unit circle; calculating, we find that $M(B) \approx 2.3692054071$.

GOAL AND APPROACH

This project will study the eigenvalues of B in relation to their distance from the unit circle. We will consider only connected quivers that do not contain any oriented cycles.

Question 1.8. *If the eigenvalues of B are not on the unit circle, how many are not on the unit circle, and how close are they to the unit circle?*

This ties directly into *Lehmer's Conjecture* or *Lehmer's Question*:

Question 1.9. *Does there exist some $C > 1$ such that for every polynomial $P(x) \in \mathbb{Z}[x]$, $M(P) = 1$ or $M(P) > C$ [2]?*

Lehmer conjectured that $C \approx 1.1762808$, given by $M(P)$ for $P(x) = x^{10} + x^9 - x^7 - x^6 - x^5 - x^4 - x^3 + x + 1$ [2]. Our approach will use the properties of E to derive properties of B and B's eigenvalues. Currently, three questions are of particular interest:

Question 1.10. *Which polynomials P can be written as the characteristic polynomial of a matrix B such that $B = -E^T E^{-1}$ for some Euler Matrix E of a quiver?*

Question 1.11. *Given two quivers Q and Q', if Q is a subgraph of Q', is $M(Q) \leq (Q')$?*

Question 1.12. *Does there exist a set of quivers $F = \{F_1, F_2, \ldots F_m\}$ such that for all i, $M(F_i) > 1$ and any quiver Q with $M(Q) > 1$ contains some F_i as a subgraph?*

The idea is hopefully to show that any polynomial that could have Mahler Measure smaller than Lehmer's polynomial could be represented as the characteristic polynomial of some B. Then, if all quivers with $M(Q) > 1$ have a subgraph F_i with $M(F_i) > 1$, and the set F has some minimum $M(F_j)$, then that would show that all polynomials have Mahler Measure greater than ≈ 1.1762808.

BACKGROUND

Lehmer first introduced his question in the 1933 paper *Factorization of Certain Cyclotomic Functions* [3]. The initial paper was motivated by an approach for finding large primes [4]. A lemma of Kronecker implies that if $M(P) = 1$, then P is a product of powers of x and cyclotomic polynomials [4].

Some partial results bounding Mahler measure have been established. Breusch proved in 1951 that a monic, irreducible, and nonreciprocal polynomial P then

$$M(P) \geq 1.324717\dots$$

which is the real root of $x^3 - x - 1$ [4]. Later, in 1979, Dobrowolski showed that for a monic, irreducible, and non-cyclotomic P,

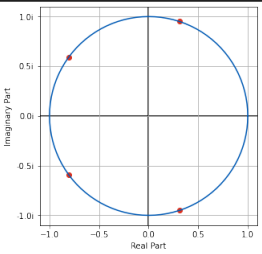$$M(P) > 1 + c \left(\frac{\log \log d}{\log d} \right)^3$$

for some constant c [4].

Several computational searches have been carried out in addition to the above approaches. None have yielded a counterexample to Lehmer's question [2]. Mossinghoff carried out a search for all polynomials of degree at most 24 with Mahler measure less than 1.3; while the specific algorithm used applied only to even degrees, he was able to find 48 such polynomials of degree 22 and 46 of degree 24 [2]. He was additionally able to use several other algorithms to extend the search calculations to more degrees, and was able to find a limit point for measures near 1.309 [2].

For more information on the work around Lehmer's question, see the cited works *Mahler Measure of Polynomials* and *Polynomials with Small Mahler Measure*.

EXAMPLES

This section contains a table of quivers, their matrices B, and plots of the eigenvalues of B in the complex plane. First, we have the simply laced Dynkin diagrams.

Graph	$B = -E^T E^{-1}$	Plot of Eigenvalues of B
	$\begin{pmatrix} -1 & -1 & -1 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	

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

Table 1: ADE-quivers and roots of B

Next we have the extended Dynkin diagrams.

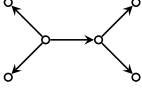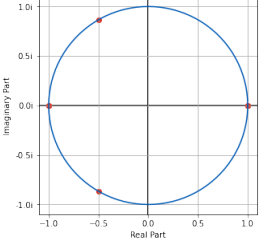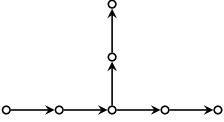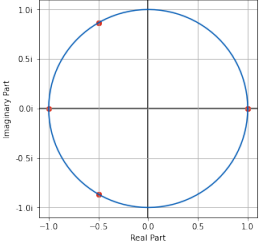| Graph	$B = -E^T E^{-1}$	Plot of Eigenvalues of B
	$$\begin{pmatrix} -1 & -1 & -1 & -1 & -2 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 2 & 2 \end{pmatrix}$$	
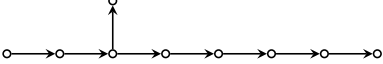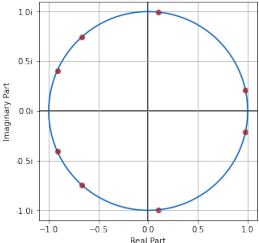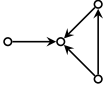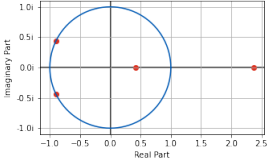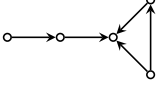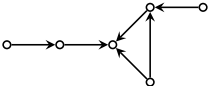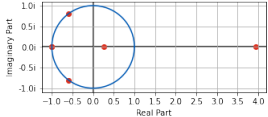
$$\begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

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

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

Table 2: Extended ADE-quivers and roots of B

Finally, we show an example of a sequence of quivers such that the quiver in each row is a subquiver of the following rows.

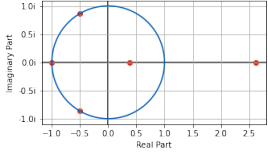| Graph	$B = -E^T E^{-1}$	Plot of Eigenvalues of B
	$\begin{pmatrix} -1 & -1 & 0 & 0 \\ 1 & 3 & 2 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & -2 & -1 & -1 \end{pmatrix}$	
	$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 3 & 2 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & -2 & -1 & -1 & 0 \\ -1 & -1 & 0 & 0 & -1 \end{pmatrix}$	
	$\begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 3 & 2 & 1 & 0 & 0 \\ 0 & 3 & 2 & 1 & 1 & 1 \\ 0 & -2 & -1 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 0 \\ 0 & -1 & -1 & 0 & 0 & -1 \end{pmatrix}$	
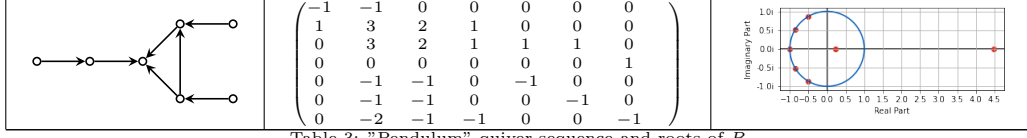
Table 3: "Pendulum" quiver sequence and roots of B

WORKS IN PROGRESS

This section contains some current remarks and propositions that have not been categorized into the rest of the paper.

Proposition 1.13. *Any polyonomial with common factor $a_d > 1$ cannot have $M(P)$ less than the Salem number.*

Proof. Consider some $P(x) = a_d \prod_{i=1}^{n}(x - \alpha_i)$ where $a_d > 1$. Since P has coefficients in the integers, a_d is an integer. Then $a_d \geq 2$, and by the definition of Mahler measure

$$M(P) = |a_d| \prod_{i=1}^{n} \max\{1, |\alpha_i|\}$$

so $M(P) \geq 2$. $\qquad\square$

As a result of this proposition, we can see that we need only to concern ourselves with polynomials that have $a_d = 1$. This is good news, as the characteristic polynomials of B do in fact have $a_d = 1$ (I think I showed this earlier).

We can also start to bound the entries of E^{-1} based on facts about triangular matrices:

Proposition 1.14. *For a matrix E of a quiver with no self-loops, the diagonal entries of E^T are 1 and the diagonal entries of E^{-1} are ± 1.*

Proof. Since we know that $\det E = 1$, E is triangular, and for a quiver with no self-loops, the diagonal entries of E are 1. So the diagonal entries of E^T are also 1. Since we know that E is triangular, we know that E^{-1} is also triangular, and since we know that $\det E^{-1} = 1$, the diagonal entries of E^{-1} must have product 1. Furthermore, since E is defined as having integer entries and has determinant 1, E^{-1} has integer entries, which means that the diagonal values of E^{-1} are ± 1. $\quad\square$

REFERENCES

[1] Derksen, H., Weyman, J., *An Introduction to Quiver Representations*, American Mathematical Society, Providence, Graduate Studies in Mathematics. **52** (2017), 1-33.

[2] Mossinghoff, M. J. *Polynomials with Small Mahler Measure*, Mathematics of Computation. **67** (1998), 1697-1705.

[3] Lehmer, D. H. *Factorization of Certain Cyclotomic Functions*, Annals of Mathematics. **34** (1933), 461-479.

[4] Lalin, Matilde N. *Mahler Measure of Polynomials* Postdoctoral Seminar, Mathematical Sciences Research Institute. (2006), 1-2.

Appendix A. Source Code

This section contains the code used to generate matrices and eigenvalue plots.

```python
# -*- coding: utf-8 -*-
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib.ticker import FormatStrFormatter
import math
import networkx as nx


# This file contains source code for explorations of Ausland-Reiter transform mat
# Example quivers are defined first, with functions to perform the necessary calc

# Define tolerance to round numbers to, may be used in np.around to round
# floating point errors to 0.
MAX_DEC = 10


###########################################################################
# EXAMPLES
# These are just a bunch of data structures that I made
# as example quivers to study.
# Notice that each quiver is technically a dictionary, and
# contains keys for:
#   name (string) - a human-readable name for the quiver
#   vertices (list of int) - list of vertices; this is just
#                            integers 0 through |V|-1 where
#                            |V| is the number of vertices.
#   edges (list of int 2-tuples) - edges in the format
#                            (tail, head).
###########################################################################

# SOME DYNKIN TYPE QUIVERS
a3 = {
    "vertices": range(3),
    "edges":[(0, 1), (1, 2)],
    "name": "A_3"
}

a4 = {
    "vertices": range(4),
    "edges": [(0, 1), (1, 2), (2, 3)],
    "name": "A_4"
}
```

```
a5 = {
    "vertices": range(5),
    "edges": [(0, 1), (1, 2), (2, 3), (3, 4)],
    "name": "A_5"
}

a6 = {
    "vertices": range(6),
    "edges": [(0, 1), (1, 2), (2, 3), (3, 4), (4, 5)],
    "name": "A_6"
}

d4 = {
    "vertices": range(4),
    "edges": [(0,1), (1, 2), (2, 4), (2, 3)],
    "name": "D_4"
}

d5 = {
    "vertices": range(5),
    "edges": [(0, 1), (1, 2), (2, 3), (2, 4)],
    "name": "D_5"
}

d6 = {
    "vertices": range(6),
    "edges": [(0, 1), (1, 2), (2, 3), (3, 4), (3, 5)],
    "name": "D_6"
}

d7 = {
    "vertices": range(7),
    "edges": [(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (4, 6)],
    "name": "D_7"
}

e6 = {
    "vertices": range(6),
    "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (4, 5)],
    "name": "E_6"
}

e7 = {
    "vertices": range(7),
    "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (4, 5), (5, 6)],
```

```
    "name": "E_7"
}

e8 = {
    "vertices": range(8),
    "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (4, 5), (5, 6), (6, 7)],
    "name": "E_8"
}

e9 = {
    "vertices": range(9),
    "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (4, 5), (5, 6), (6, 7), (7, 8)],
    "name": "E_9"
}

e10 ={
    "vertices": range(10),
    "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8,
    "name": "E_10"
}


e11 ={
    "vertices": range(11),
    "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8,
    "name": "E_11"
}

e12 ={
    "vertices": range(12),
    "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8,
    "name": "E_12"
}

e13 ={
    "vertices": range(13),
    "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8,
    "name": "E_13"
}

# OTHER EXAMPLES
ex = {
    "name": "Ex",
    "vertices": range(15),
    "edges": [(0, 1), (2, 1), (1, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8,
```

```
}

# Complete graph; this turns out to have rather boring eigenvals.
complete4 = {
    "name": "Complete_Order_4",
    "vertices": range(4),
    "edges": [(0, 1), (0, 2), (0, 3), (1, 0), (1, 2), (1, 3), (2, 0), (2, 1), (2,
}

# A cycle of length 4, but not an oriented cycle. This is A_3 Extended.
ac_4cycle = {
    "name": "Acyclic_Weakly_Connected_'Cycle'",
    "vertices": range(4),
    "edges": [(0, 1), (1, 2), (2, 3), (0, 3)]
}

# A "pendulum"-looking graph.
ac_pendulum = {
    "name": "Acyclic_Pendulum",
    "vertices": range(4),
    "edges": [(0, 1), (2, 1), (3, 1), (3, 2)]
}

# Variant on A_4
double_a4 = {
    "name": "A4_with_bidirectional_edges",
    "vertices": range(4),
    "edges": [(0, 1), (1, 2), (2, 3), (3, 2), (2, 1), (1, 0)]
}

# I drew this one that looked a little bit like the letter "M"
m4 = {
    "name": "M-shape_Acyclic",
    "vertices": range(4),
    "edges": [(0, 1), (0, 2), (2, 3), (1, 3)]
}

# "Pendulum" shape, but longer.
big_pendulum = {
    "name": "Acyclic_Pendulum",
    "vertices": range(7),
    "edges": [(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (4, 6), (5, 6)]
}

# "Pendulum" shape, but the "bottom" is wider.
```

```python
wide_pendulum = {
    "name": "Wide_Pendulum",
    "vertices": range(6),
    "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5)]
}

# A_7 extended, being sure not to have oriented cycle.
a7_ext_ac = {
    "name": "A_7_Extended_with_Acyclic_Orientation",
    "vertices": range(8),
    "edges": [(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (0, 7)]
}


############################################################################
# FUNCTIONS
# These functions do a bunch of calculations. See each
# function's documentation for specific details.
############################################################################

def head(edge):
    """Returns the head of the edge given as a 2-tuple.

    Arguments:
        edge (2-tuple): an edge specified as (out, in)

    Returns:
        head (int) - the head of the edge.
    """
    return edge[1]

def tail(edge):
    """Returns the tail of the edge given as a 2-tuple.

    Arguments:
        edge (2-tuple): an edge specified as (out, in)

    Returns:
        tail (int) - the tail of the edge.
    """
    return edge[0]

def E(q):
    """Transforms a quiver object into its corresponding Euler matrix.

    Arguments:
```

```
        q (quiver) = a quiver with vertices (list of int) and edges (list of 2-tuples

    Returns:
        euler_matrix (numpy matrix) - the Euler matrix of q.
    """

    # We're going to use a 2D python array at first.
    euler_array = [[0 for x in range(len(q["vertices"]))] for y in range(len(q["ver
    for out_vertex in range(np.size(q["vertices"])):
        for in_vertex in range(np.size(q["vertices"])):
            # Set Kronecker symbol
            kronecker = 0
            if out_vertex == in_vertex:
                kronecker = 1

            # iterate through edges and find edges with appropriate head and tail.
            edges = [edge for edge in q["edges"] if head(edge) == in_vertex and tail(ed
            euler_array[out_vertex][in_vertex] = kronecker - len(edges)

    return np.matrix(euler_array)

def B(q):
    """Calculates the matrix B from the given quiver.

    Arguments:
        q (quiver) - the quiver to calculate the transformation matrix B from.

    Returns:
        B (numpy matrix)
    """
    e = E(q)
    et = e.transpose()
    e_inv = np.linalg.inv(e)
    b = -1 * et * e_inv

    return b

# Overlaying roots of unity over the eigenvalues is occasionally illuminating.
def nth_roots_unity (n):
    """Calculates the nth roots of unity.

    Arguments:
        n (int) - some positive integer to calculate the roots of unity of.
    Returns:
        roots (numpy array) - the nth roots of unity.
```

```python
    """
    p = [1] + [0] * (n − 1) + [−1]
    roots = np.roots(p)
    return roots

def mahler_measure_from_eigens(eigens):
    """Calculates the mahler measure of a characteristic polynomial that has
    the given eigenvalues as roots.

    Arguments:
        eigens (numpy array) − 1D array containing the eigenvalues, which may be
        complex numbers.

    Returns:
        measure (real number) − the Mahler measure of the polynomial given by
        (x − eigens[0])(x − eigens[1])...(x−eigens[np.size(eigens) − 1]).
    """

    mahler = np.prod([np.max([1, np.absolute(x)]) for x in np.around(eigens, MAX_DE
    return mahler

def eigens_from_quiver(q):
    """Given a quiver q, returns the eigenvalues of A = (E^T)^−1 where E is the Eul
    quiver.

    Arguments:
        q (dictionary) − a dictionary with keys vertices (list of int) and edges (lis

    Returns:
        eigens (list of complex numbers) − the eigenvalues of the matrix A.
    """

    eigens = np.linalg.eigvals(B(q))

    return eigens

def format_subplot(ax, eigens):
    """Given a matplotlib ax object, call some formatting methods.

    Arguments:
        ax (matplotlib.plot.figure.subplot) − the subplot object.
        eigens (numpy array) − the eigenvalues being plotted, used to set axis bounds
    """

    # Create horizontal lines to denote the x + 0i and 0 + yi lines for clarity.
```

```python
    ax.axhline(color='black', zorder=-100001)
    ax.axvline(color='black', zorder=-100000)
    # Set aspect ratio to 1 to avoid weirdness
    ax.set_aspect(1.0/ax.get_data_ratio())
    #Label axes
    ax.set_xlabel('Real Part')
    ax.set_ylabel('Imaginary Part')
    # Set axis ticks; notice that the stopping point is ceil(max).1 for both ranges
    # Otherwise, it will not render the final tick properly.
    e_max_r = np.max([eigen.real for eigen in eigens])
    e_min_r = np.min([eigen.real for eigen in eigens])
    e_max_i = np.max([eigen.imag for eigen in eigens])
    e_min_i = np.min([eigen.imag for eigen in eigens])

    ax.yaxis.set_ticks(np.arange(math.floor(e_min_i), math.ceil(e_max_i) + 0.1, 0.5
    ax.xaxis.set_ticks(np.arange(math.floor(e_min_r), math.ceil(e_max_r) + 0.1, 0.5
    ax.yaxis.set_major_formatter(FormatStrFormatter('%si'))
    ax.yaxis.grid(which='both')
    ax.xaxis.grid(which='both')
    ax.grid(True)

def plot_eigenvals(eigens, ax):
    """ Given the eigenvalues and subplot, plot the eigenvalues on the subplot.

    Arguments:
        eigens (list of complex numbers) - the eigenvalues to plot.
        ax (matplotlib subplot) - subplot to graph eigenvals on.
    """

    X = np.around([x.real for x in eigens], MAX_DEC)
    Y = np.around([x.imag for x in eigens], MAX_DEC)
    ax.scatter(X,Y, color='#DC3220', zorder=-10)

    # Also plot the unit circle
    #plot unit circle
    t = np.linspace(0,2*math.pi,101)
    ax.plot(np.cos(t), np.sin(t), color='#005AB5')

def plot_quiver_graph(q, ax):
    """Plots the given quiver using NetworkX MultiDiGraph.

    Arguments:
        q (quiver) - the quiver to plot.
        ax (axis) - the matplotlib axis to draw on.
    """
```

```python
  G = nx.MultiDiGraph()
  G.add_edges_from(q["edges"])

  nx.draw_networkx(G, ax=ax)

def plot_quivers_eigenvals(quiver_list):
  """Plots the eigenvalues of A for every quiver in the list and draws the graph

  Arguments:
    quiver_list (list of quiver) - the quivers to plot.
  """
  fig, axs = plt.subplots(len(quiver_list), 1, figsize=(10, 5))
  for n in range(len(axs)):
    # Get the eigenvals
    q = quiver_list[n]
    eigens = eigens_from_quiver(q)
    eigen_ax = axs[n]
    format_subplot(eigen_ax, q)
    plot_eigenvals(eigens, eigen_ax)

  # Margin is a bit too tight, so give each plot some more room.
  plt.subplots_adjust(bottom=-2)
  plt.show()

def plot_quiver_eigenvals(quiver):
  """Plots the eigenvalues of the Auslander-Reiten transform matrix B for a singl

  Arguments:
    quiver (quiver) - the quiver to calculate B for and plot eigenvalues of.
  """
  fig, ax = plt.subplots(figsize=(5, 5))
  eigens = eigens_from_quiver(quiver)
  format_subplot(ax, eigens)
  plot_eigenvals(eigens, ax)

  plt.show()

def to_pmatrix(mat):
  """Converts some matrix into a string that can be pasted into a LaTeX document.

  Returns:
    bmat (string) = a string of the form a11 & a12 & ... \\ a21 & a22 & ... \\
    for pasting into LaTeX.
  """
  bmat = ''
```

```python
    np_mat = np.array(mat)
    for row in np_mat:
        for n in row:
            bmat += str(int(n))
            bmat += ' & '
        bmat += '\\\\ '
    return bmat

# This is sort of the "final" function that will display everything nicely.
def get_nums(quiver):
    """Prints some information for the given quiver. Includes the Euler Matrix
    and the matrix B in a form suitable for pasting into a LaTeX doc, as well as
    the Mahler measure of the quiver and a plot of the eigenvalues of B.
    """

    print(to_pmatrix(E(quiver)) + "\n")
    print(to_pmatrix(B(quiver)) + "\n")
    print(eigens_from_quiver(quiver))

    print(mahler_measure_from_eigens(eigens_from_quiver(quiver)))
    plot_quiver_eigenvals(quiver)

# Use this function on whichever quiver you want to graph and get matrices for.
get_nums(e8)
```

DEPARTMENT OF MATHEMATICS, NORTHEASTERN UNIVERSITY, BOSTON, MASSACHUSETTS 02115
Email address: nichols.t@northeastern.edu