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 investigate the properties of polynomials formed by the matrix B defined on a quiver (see sections 1.1-1.5). In particular, we ask these questions:

Question 1.1. *Which polynomials P can be written as the characteristic polynomial of a matrix B for some Euler Matrix E of a quiver?*

Question 1.2. *Given two quivers Q and Q', if Q is a subgraph of Q', is $M(Q) \leq M(Q')$?*

Question 1.3. *Given a triangular matrix E, is it the case that if $E+E^T$ is positive semidefinite, then the characteristic polynomial of B has roots on the unit circle in the complex plane?*

We now introduce 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 $0 \leq i \leq n$, $h(e_{i-1}) = t(e_i)$ and $t(e_0) = u$, $h(e_n) = v$.

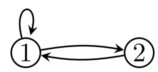
Example 1.4. Consider the quiver illustrated by the following diagram:

$$1 \to 2 \to 3 \to 4$$

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

Date: November 8, 2020.

Example 1.5. 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.

Example 1.6. Consider again the quivers in Examples 1.1 and 1.2. Observe that the quiver in Example 1.1 does not contain an oriented cycle, as there is no path from any node back to itself. However, the quiver in Example 1.2 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

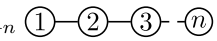
$$\delta_{i,j} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}$$

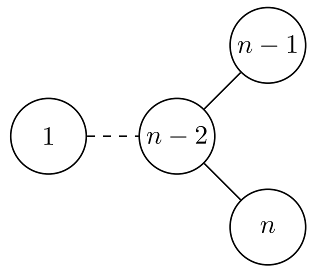
1.3. **Ausland-Reiter Transform.** This project studies the matrix product defined by $B = -E^T E^{-1}$, where E is the Euler Matrix for some quiver Q. The matrix B is used in the Ausland-Reiter Transform [1]; however, defining the transform is currently beyond the scope of this project.

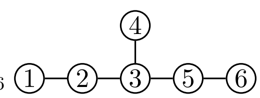
An important note is that B is defined using E^{-1}, but E is singular when its defining quiver is cyclic (TODO: find citation). Therefore, we will generally be considering only acyclic quivers.

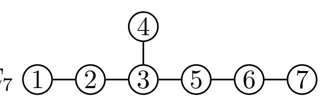
1.4. **Dynkin Diagrams.** Of particular importance in representation theory are the *Dynkin diagrams*, which are related to irreducible root systems and Lie algebras [1]. Many of the connections in which these diagrams appear are beyond the scope of this paper, but the diagrams themselves will be important,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

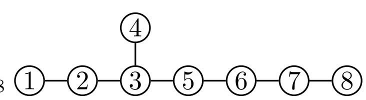
Definition 6. [1] The *simply laced Dynkin diagrams* are A_n, D_n, E_6, E_7, and E_8 [1]. Dynkin diagrams are typically drawn as undirected graphs; if we speak of a quiver as being related to a Dynkin diagram, we mean the quiver drawn also as an undirected graph.

A_n 

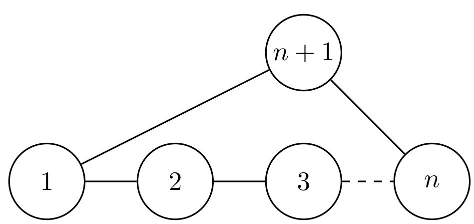
D_n 

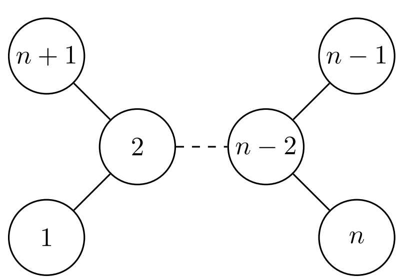
E_6 

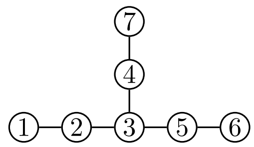
E_7 

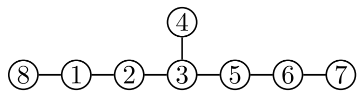
E_8 

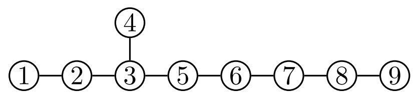
Also of interest are the *extended Dynkin diagrams* $\hat{A}_n$, $\hat{D}_n$, $\hat{E}_6$, $\hat{E}_7$, and $\hat{E}_8$.

$\hat{A}_n$ 

$\hat{D}_n$ 

$\hat{E}_6$ 

$\hat{E}_7$ 

$\hat{E}_8$ 

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.

1.5. **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 1.7. Since the matrix B has integer coefficients, its characteristic polynomial has integer coefficients, and since the characteristic polynomial is monic by definition, the leading factor $|a_d|$ will always be 1. This means that

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

We can see immediately that $M(B) \geq 1$. Additionaly, 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 1.8. *For any matrix B with integer entries,* $\det(B) \leq M(B)$.

Proof. Suppose an $n \times n$ 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 $|\det(B)| \leq |M(B)|$. ∎

Example 1.9. Consider the quiver given by

$$1 \to 2 \to 3 \to 4$$

Then

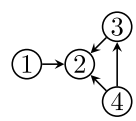
$$E = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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 1.10. Consider the quiver A_4 given by



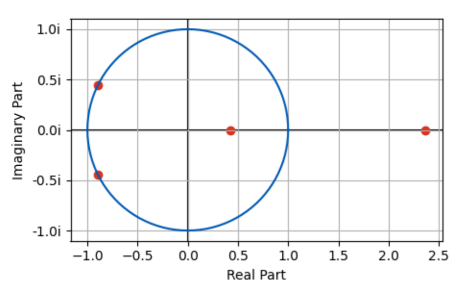
Then

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

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 $M(B) \approx 2.3692054071$.

However, it is not known if there is some $P(x)$ with $M(P) = 1 + \epsilon$ for $\epsilon > 0$. This question is referred to as *Lehmer's Conjecture* or *Lehmer's Question* [2]:

Question 1.11. *Does there exist some $C > 1$ such that for every polynomial $P(x) \in \mathbb{Z}[x]$, $M(P) = 1$ or $M(P) > C$?*

Lehmer conjectured that $C \approx 1.1762808$, given by $M(P)$ for $P(x) = x^{10} + x^9 - x^7 - x^6 - x^5 - x^4 - x^3 + x + 1$; see [2].

1.6. **Topological Ordering.** Importantly, we know that any directed acyclic graph has a *topological ordering* of vertices; see [8], wherein any edge e has $h(e) \geq t(e)$, or in other words, the vertices are numbered in such a way that the edges all go in one direction.

Example 1.12. Consider A_3 as numbered below:

$$\begin{tikzcd} 1 \arrow[r] & 2 & 3 \arrow[l] \end{tikzcd}$$

We may construct a topological ordering of this graph by re-numbering vertex 3 as vertex 2. We can then draw the graph like this:

$$\begin{tikzcd} 1 \arrow[rr, bend right] & 2 \arrow[r] & 3 \end{tikzcd}$$

Notice that now the edges all flow in the same direction. If we consider the example as an acyclic quiver, we see that arranging the vertices in a topological order means that E will be an upper triangular matrix; in particular,

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

And this can similarly be done with a reversed ordering of the vertices to obtain a lower triangular E. It follows from the definitions that any topological ordering of a quiver will lead to a E being triangular. Therefore, we will use this fact to assume that E is defined as a triangular matrix when we use it in examples and calculations. According to professor Derksen, if such an ordering can be constructed, it will not change what the eigenvalues of B are.

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. We reference the questions from the introduction.

We consider the following general question as an overall direction for studies:

Question 1.13. *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 Question*; see [2].

Question 1.14. *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?*

We can see that if we were not studying monic polynomials, we would be unable to achieve a Mahler measure smaller than 2:

Proposition 1.15. *Any non-monic polynomial $P(x)$ 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

$$M(P) = 2 \prod_{i=1}^{n} \max\{1, |\alpha_i|\}$$

so $M(P) \geq 2$. □

As a result of this proposition, we can see that we need only to concern ourselves with monic polynomials if we want to find a polynomial with smaller measure than Lehmer's. This is good news, as the characteristic polynomials of matrices are always monic by definition.

Furthermore, Professor Veliche noticed that for some root λ of a quiver, $\frac{1}{\lambda}$ appears to also be a root. The complex roots appear to lie on the unit circle, so their reciprocals are simply their conjugates, but in cases with real roots other than 1, it appears that they are indeed reciprocals. This notion corresponds with the definition of a *reciprocal* or *palindromic polynomial*, which is defined to have palindromic coefficients.

Therefore we propose the following:

Proposition 1.16. *The characteristic polynomial of B is reciprocal.*

Proof. Let $B = -E^T E^{-1}$ and E be an $n \times n$ triangular Euler matrix. Let $P(x) = \det xI - B$ Then $xI - B = xI + E^T E^{-1}$. Multiply by E on the right to obtain $(xI - B) = xE + E^T$. But since $\det E = 1$, $\det xI - B = \det xE + E^T$. We know that since $\det B = \pm 1$, $x = 0$ is not a root of $\det xI - B$, and so $\det xI - B = \det xE + E^T = 0$ means that $\det xE + E^T = x^n \det E + \frac{1}{x}E^T$. But we can see that for $x \neq 0$, $P(\frac{1}{x}) = \det E + \frac{1}{x}E^T$, so $P(x) = x^n P(\frac{1}{x})$, and therefore by [6] P is reciprocal. □

If x_i is some root of a reciprocal polynomial $P(x)$, then $\frac{1}{x_i}$ is also a root of $P(x)$, which corresponds with the observation that the roots of the characteristic polynomial of B occur in reciprocal pairs.

Question 1.17. *Given a triangular matrix E, is it the case that if $E + E^T$ is positive semidefinite, then the characteristic polynomial of B has roots on the unit circle in the complex plane?*

According to professor Derksen, this is known to be true via the classification of quiver representations; see [1]. However, we would like to demonstrate this relationship from the matrices themselves, without relying on the apparatus of quiver representation theory.

Question 1.18. *Given two quivers Q and Q', if Q is a subgraph of Q', is $M(Q) \leq M(Q')$?*

We consider an example given by adding one vertex and edge to a small quiver in the table below:

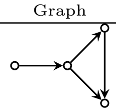
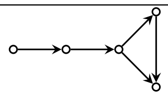
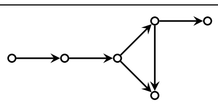
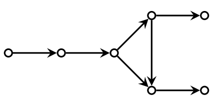
Graph	$B = -E^T E^{-1}$	Characteristic Polynomial of B	$M(B)$
	$\begin{pmatrix} -1 & -1 & -2 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 2 \\ 0 & 1 & 1 & 0 \end{pmatrix}$	$\lambda^4 - \lambda^3 - 3\lambda^2 - \lambda + 1$	2.3692054071
	$\begin{pmatrix} -1 & -1 & -1 & -1 & -2 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2 & 2 \end{pmatrix}$	$\lambda^5 - \lambda^4 - 3\lambda^3 - 3\lambda^2 - \lambda + 1$	2.618033988770564
	$\begin{pmatrix} -1 & -1 & -1 & -1 & -2 & -2 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$	$\lambda^6 - \lambda^5 - 5\lambda^4 - 7\lambda^3 - 5\lambda^2 - \lambda + 1$	3.3014904324145147
	$\begin{pmatrix} -1 & -1 & -1 & -1 & -2 & -2 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$	$\lambda^7 - \lambda^6 - 7\lambda^5 - 13\lambda^4 - 13\lambda^3 - 7\lambda^2 - \lambda + 1$	3.899908930943441

Table 1: "Pendulum" quiver sequence and characteristic polynomials of B

Notice that the Mahler measure of each quiver is increasing as more edges and vertices are added. We would like to show that this, in fact, must be the case, and the addition of any edges or vertices to a quiver cannot lead to a smaller Mahler measure than the original quiver.

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; see [4].

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

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

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; see [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; see [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.

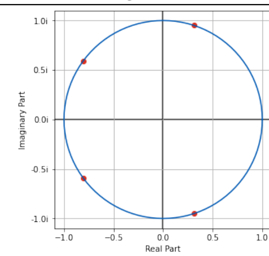
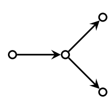
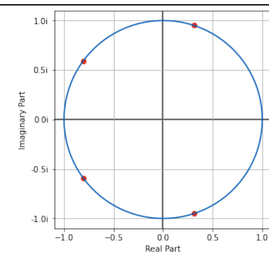
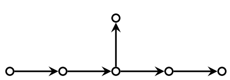
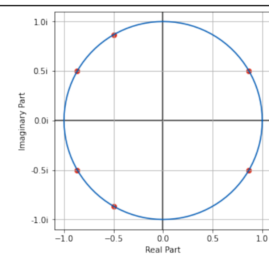
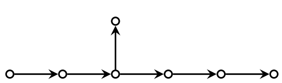
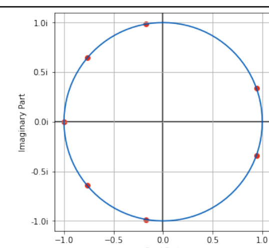
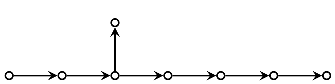
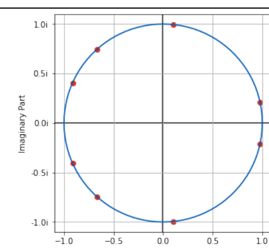
Graph	$B = -E^T E^{-1}$	Plot of Eigenvalues of B
	$\begin{pmatrix} -1 & -1 & -1 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	
	$\begin{pmatrix} -1 & -1 & -1 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	
	$\begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1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

Table 2: ADE-quivers and roots of B

Next we have the extended Dynkin diagrams.

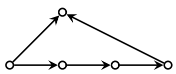
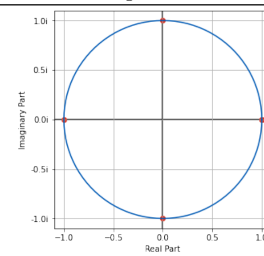
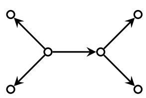
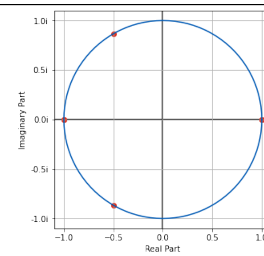
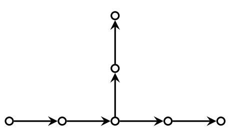
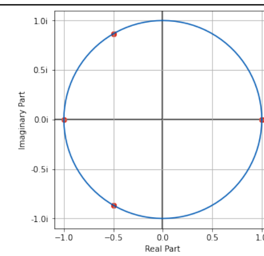
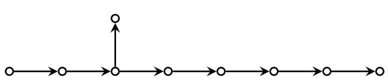
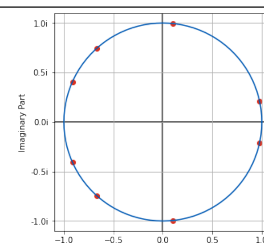
Graph	$B=-E^TE^{-1}$	Plot of Eigenvalues of B
	$\begin{pmatrix} -1 & -1 & -1 & -1 & -2 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 2 & 2 \end{pmatrix}$	
	$\begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 &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

Table 3: 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	$B=-E^TE^{-1}$	Plot of Eigenvalues of B

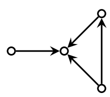
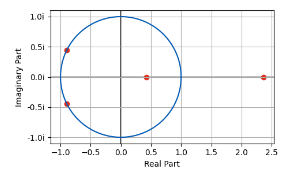
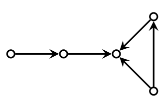
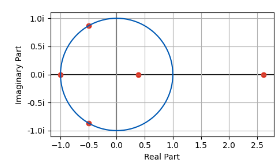
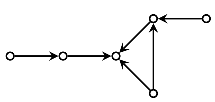
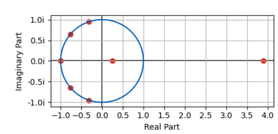
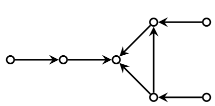
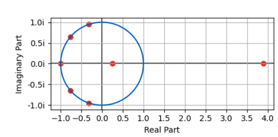
	$\begin{pmatrix} -1 & -1 & 0 & 0 \\ 1 & 3 & 2 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & -2 & -1 & -1 \end{pmatrix}$	
	$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 3 & 2 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & -2 & -1 & -1 & 0 \\ -1 & -1 & 0 & 0 & -1 \end{pmatrix}$	
	$\begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 3 & 2 &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

Table 4: "Pendulum" quiver sequence and roots of B

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.

[5] Bernstein, Dennis S. *Matrix Mathematics* , Princeton University Press. (2005), 44.

[6] Sun, H., Wang, Y., Zhang, H. X., *Polynomials with Palindromic and Unimodal Coefficients*, Acta Mathematica Sinica, English Series. **31** (2015), 565-575.

[7] Stankov, Dragan, *The number of unimodular roots of some reciprocal polynomials*, Reports. Mathematics, **358** (2020), 159-168.

[8] Sedgewick, R., Wayne, K., *Algorithms: Fourth Edition, Part II*, Addison-Wesley, Boston. (2014), 574-578.

Appendix A. Source Code

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

```
# -*- 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
import sys

# This file contains source code for explorations of Ausland-Reiter transform matrices B for small quviers.
# Example quivers are defined first, with functions to perform the necessary calculations following.

# 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, 9)],
    "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, 9), (9, 10)],
    "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, 9), (9, 10), (10, 11)],
    "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, 9), (9, 10), (10, 11), (11, 1
    "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, 9), (9, 10), (10, 11), (10, 1
}

# 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, 3), (3, 0), (3, 1), (3, 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), (1, 2), (1, 3), (3, 2)]
}

ac_pendulum_2 = {
    "name": "Acyclic Pendulum",
    "vertices": range(5),
    "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (3, 4)]
}

ac_pendulum_3 = {
    "name": "Acyclic Pendulum",
    "vertices": range(6),
    "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (3, 4), (4, 5)]
}

ac_pendulum_4 = {
    "name": "Acyclic Pendulum",
    "vertices": range(7),
    "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (3, 4), (4, 5), (3, 6)]
}

# 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.
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)]
}

ac_pendulum_2_U_a2 = {
  "name": "Pendulum graphs union",
  "vertices": range(7),
```

```
  "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (3, 4), (3, 5), (5, 6)]
}

ac_pendulum_2_U_d4 = {
  "name": "Pendulum_graphs_union",
  "vertices": range(8),
  "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (3, 4), (3, 5), (5, 6), (5, 7)]
}

ac_pendulum_2_U_ac_pendulum = {
  "name": "Pendulum_graphs_union",
  "vertices": range(8),
  "edges": [(0, 1), (1, 2), (2, 3), (2, 4), (3, 4), (3, 5), (5, 6), (6, 7), (5, 7)]
}

tri = {
  "name": "Tri",
  "vertices": range(3),
  "edges": [(0, 1) , (1, 2), (0, 2)]
}

quad = {
  "name": "Quad",
  "vertices": range(4),
  "edges": [(0, 1) , (1, 2), (0, 2), (0, 4)]
}

############################################################
# 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["vertices"]))]
  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(edge) == out_vertex]
        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

# Calculate a string representing the characteristic polynomial of a matrix.
def tex_char_poly(A, var="\\lambda"):
    """Returns a LaTeX string for the characteristic polynomial of a given matrix
    A

    Arguments:
      A (matrix) - the matrix to calculate the characteristic polynomial of.
      var (string, optional) - string to use as the variable name.

    Returns:
      poly_string (string) - the characteristic polynomial in LaTeX form.
    """

    coeffs = [int(np.around(x, MAX_DEC)) for x in np.poly(A)]
    N = len(coeffs)
    latex_string = ""

    for i in range(0, N):
      # Can just round this to integers, since we know it will be.
      c = coeffs[i]
      v_power = var + "^{" + str(N - i - 1) + "}"

      # Set the sign for th next term. We do this even if this term will
      # not be displayed.
      op = "+" if i != 0 else ""
      if (c < 0):
          op = "-"
      # Need to handle the constant term differently
      if (i != N - 1):
        if (c != 0):
          # Sign is handled, so take absolute value of c
          c = abs(c)
          # For non-constant terms, omit a coefficient of 1 or -1
          if (c == 1):
            c = ""
          if (i == N - 2):
            # If power is 1, don't show it.
            v_power = var
          # Determine if next character should have pos/neg connector.
          latex_string += op + " " + str(c) + v_power + " "
      else:
        if (c != 0):
          latex_string += op + " " + str(c)

    # I think if the constant term is 0 we'd have a trailing +; just in case,
    # remove it before returning.
    return latex_string.strip("+ -")
```

```
# 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.
  """
  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_DEC)])
  return mahler

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

  Arguments:
    q (dictionary) - a dictionary with keys vertices (list of int) and edges (list of 2-int tuples).

  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.
  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_quivers_eigenvals(quiver_list):
  """Plots the eigenvalues of A for every quiver in the list and draws the graph next to it.

  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 single quiver.

  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 = ''
  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, var="\\lambda"):
  """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(tex_char_poly(B(quiver), var) + "\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.
try:
    var = sys.argv[2]
except IndexError:
    var = "\\lambda"
get_nums(globals()[sys.argv[1]], var)
```

Department of Mathematics, Northeastern University, Boston, Massachusetts 02115
Email address: nichols.t@northeastern.edu