ABSTRACT

The MP Algorithm and Its Applications

Christopher Bailey, Ph.D.
Department of Mathematical Sciences
Northern Illinois University, 2013
Dr. Yoopyo Hong, Director

The minimal polynomial of a matrix contains some very basic information on the matrix (e.g., spectral structure). Consequently, the minimal polynomial of a given matrix provides useful and important knowledge in analysing matrices that arise in applications. In practice, however, it is important to know not only the minimal polynomial itself but also the mechanism of obtaining the minimal polynomial. We present an algorithmic method for computing the minimal polynomial of any square matrix. Since the method uses basic operations, it is algebraically and theoretically simple to apply. The method allows us to compute the exact minimal polynomial of any reasonably small size matrix. In general, it is not possible in any conventional method for obtaining the minimal polynomial. In certain cases, the simplicity of the method enables to show some desirable properties of the matrix.

NORTHERN ILLINOIS UNIVERSITY
DE KALB, ILLINOIS

JUNE 2013

THE MP ALGORITHM AND ITS APPLICATIONS

BY

CHRISTOPHER BAILEY

A DISSERTATION PAPER SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE
DOCTOR OF PHILOSOPHY

DEPARTMENT OF MATHEMATICAL SCIENCES

Doctoral Director:
Dr. Yoopyo Hong

ACKNOWLEDGMENTS

I would like to express my deep gratitude to Professor Hong, my advisor, for his patient guidance, enthusiastic encouragement, and useful critiques of this thesis. I would also like to thank my wife, Abigail, whose loving encouragement, careful editing, and useful suggestions allowed me to complete this dissertation.

Thank you to my parents, George and Alice, for their support throughout this endeavor. I also thank my external reviewer and everyone on my committee for their willingness to give their time so generously.

DEDICATION

For my daughter, Amelia. One in seven billion.

TABLE OF CONTENTS

The following definitions and notations will be used throughout this dissertation.

$M_n(\mathbb{C})$	the set of all $n \times n$ matrices with complex entries
I_n	identity matrix in $M_n(\mathbb{C})$
$U_n(\mathbb{C})$	the set of all $n \times n$ unitary matrices
$\mathbb{C}^n$	$\{(a_1, \ldots, a_n)^T \in M_{1\times n}(\mathbb{C}) \mid a_i \in \mathbb{C}\}$
$\mathbb{C}_n$	$\{(a_1, \ldots, a_n) \in M_{n\times 1} \mid a_i \in \mathbb{C}\}$
$H_n(\mathbb{C})$	the set of all $n \times n$ Hermitian matrices
Π_n	the set of all $n \times n$ permutation matrices
$p_A(t)$	the characteristic polynomial of a matrix $A \in M_n(\mathbb{C})$
$q_A(t)$	the minimal polynomial of a matrix $A \in M_n(\mathbb{C})$
$tr(A)$	the trace of a matrix $A \in M_n(\mathbb{C})$
$det(A)$	the determinant of matrix $A \in M_n(\mathbb{C})$
$diag(\lambda_1, \ldots, \lambda_n)$	the diagonal matrix with diagonal entries λ_i, for $i = 1, \ldots, n$
$A \otimes B$	The Kronecker product of matrices $A \in M_n(\mathbb{C})$ and $B \in M_n(\mathbb{C})$
$vec(A)$	The vec of the matrix $A \in M_n(\mathbb{C})$
$D_n(\mathbb{C})$	the set of all $n \times n$ diagonal matrices
$RD_n(\mathbb{C})$	the set of all $n \times n$ real diagonalizable matrices
$\sigma(A)$	the set of all eigenvalues of a matrix $A \in M_n(\mathbb{C})$
$deg(p(t))$	the degree of a polynomial $p(t)$

CHAPTER 1

Introduction

1.1 Overview of dissertation

This dissertation develops an algorithm to compute the minimal polynomial of any square matrix with exact arithmetics and demonstrates how to apply the algorithm to a matrix. Chapter 1 introduces the minimal polynomial with a discussion on existing methods for computing it. Then, applications of the minimal polynomial are provided. After a brief review of some existing methods for computing the minimal polynomial, Chapter 2 introduces and develops a new algorithm to compute the minimal polynomial for any square matrix. The algorithm is then modified to make it easier to implement. Chapter 2 concludes with examples that illustrate how the algorithm works. Chapter 3 applies the new algorithm to matrices of special structure. In particular, the algorithm is applied to the class of lower Hessenberg matrices where the algorithm simplifies when it is applied. In addition to applying the algorithm to matrices of special structure, some interesting consequences of the algorithm are noted in Chapter 3. A further application of the new algorithm is applied to a new class of real diagonalizable matrices that are defined in Chapter 4. A systematic approach is taken for determining whether a matrix is or is not real diagonalizable. Some observations of this new class of matrices are made, including how the algorithm can be used to compute its minimal polynomial.

1.2 The minimal polynomial

For any polynomial $p(t) = t^k + a_{k-1}t^{k-1} + \cdots + a_1 t + a_0$ and any $A \in M_n(\mathbb{C})$ the matrix polynomial $p(A) = A^k + a_{k-1}A^{k-1} + \cdots + a_1 A + a_0 I$ is well defined. When $p(A) = 0 \in M_n(\mathbb{C})$, the polynomial $p(t)$ is said to annihilate matrix A. The minimal polynomial of matrix $A \in M_n(\mathbb{C})$ is a monic polynomial of minimal degree that annihilates the matrix A, and it is denoted by $q_A(t)$. Any matrix $A \in M_n(\mathbb{C})$ has an annihilating polynomial of finite degree, which we show here. It can easily be seen that the set of all n-by-n complex matrices forms a vector space over the field of complex numbers under the usual operations on matrices. Let $E_{ij} \in M_n(\mathbb{C})$ be the matrix that has a one in the (i,j)-th entry and zero in all other entries. First, it is clear that the set $\{E_{ij}\}$ is linearly independent and any matrix $A = [a_{ij}] \in M_n(\mathbb{C})$ can be written uniquely as $A = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}E_{ij}$. This shows that the set $\{E_{ij}\}$ forms a basis for $M_n(\mathbb{C})$. Consequently, the vector space $M_n(\mathbb{C})$ has dimension n^2. Since the set $D = \left\{A^0, A, A^2, \ldots, A^{n^2}\right\}$ contains n^2+1 elements, D must be a linearly dependent set for any $A \in M_n(\mathbb{C})$. Thus, there is a linear combination $\alpha_0 I + \alpha_1 A + \cdots + \alpha_{n^2}A^{n^2}$, $\alpha_i \in \mathbb{C}, i = 1, \ldots, n^2$ are not all zero such that $\sum_{i=0}^{n^2} \alpha_i A^i = 0$. In other words, the polynomial $p(t) = \sum_{i=0}^{n^2} \alpha_i t^i$ is an annihilating polynomial of A. Therefore, an annihilating polynomial exists for any matrix $A \in M_n(\mathbb{C})$. This means that a minimal polynomial exists for any $A \in M_n(\mathbb{C})$. Not only does the minimal polynomial exit, but it is unique [12, p. 642]. The Cayley-Hamilton theorem [5, Theorem 2, p. 83] asserts that the degree of the minimal polynomial is at most n. In contrast to the simplicity of knowing that the minimal polynomial exists, computing the minimal polynomial is an entirely different matter.

There are several reasons for wanting to know the minimal polynomial of a given matrix $A \in M_n(\mathbb{C})$. One important reason is that it contains information about all of the distinct eigenvalues of the matrix A. This is significant because the diagonalizablity of a matrix A is completely determined by the minimal polynomial, since a matrix $A \in M_n(\mathbb{C})$ is diagonalizable if and only if the minimal polynomial factors into distinct linear factors.

Another good use of the minimal polynomial is that it provides a way to write a matrix polynomial in the simplest possible form. To see how, suppose that $q_A(t)$ is the minimal polynomial of a given matrix $A \in M_n(\mathbb{C})$. For any polynomial $p_m(t)$ with $m \geq deg(q_A(t))$, we may apply the Euclidean algorithm by dividing the polynomial $p_m(t)$ by the minimal polynomial $q_A(t)$ to obtain $p_m(t) = h(t)q_A(t) + r(t)$, where $h(t)$ and $r(t)$ are the quotient and remainder, respectively, such that $deg(r(t)) < deg(q_A(t))$. This means that $p_m(A) = h(A) \cdot q_A(A) + r(A) = r(A)$ since $q_A(A) = 0$. In particular $p_m(A) = r(A)$, which reduces the original matrix polynomial $p_m(t)$ of degree m to an equivalent matrix polynomial of degree one less than the degree of the minimal polynomial. To see a specific case of this, consider the polynomial $p_4(t) = t^4 + 3t^3 + t^2 + 3$, and suppose that the matrix $A \in M_n(\mathbb{C})$ has minimal polynomial $q_A(t) = t^2 + 1$. When we divide $p_4(t)$ by the minimal polynomial the result is $p_4(t) = t^4 + 3t^3 + t^2 + 3 = (t^2 + 3t)q_A(t) + (-3t + 3)$, and therefore $p_4(A) = A^4 + 3A^3 + A^2 + 3I = -3A + 3I$. This reduction process is an important tool in the power series expansions of a function on matrices.

The minimal polynomial of a matrix also provides us with a tool to compute the inverse of a given invertible matrix $A \in M_n(\mathbb{C})$. Suppose that the minimal polynomial $q_A(t) = t^k + a_{k-1}t^{k-1} + \cdots + a_1 t + a_0$ for an invertible matrix $A \in M_n(\mathbb{C})$. Then $A^k + a_{k-1}A^{k-1} + \cdots + a_1 A + a_0 I = 0$. We may rewrite the equation as $A(A^{k-1} + a_{k-1}A^{k-2} + \cdots + a_1 I) = -a_0 I$, and because $a_0 = det(A) \neq 0$, we have $A(\frac{-1}{a_0}(A^{k-1} + a_{k-1}A^{k-2} + \cdots + a_1 I)) = I$. Thus the inverse of the matrix $A \in M_n(\mathbb{C})$

is given by $A^{-1} = \frac{-1}{a_0}(A^{k-1} + a_{k-1}A^{k-2} + \cdots + a_1 I)$ which is a polynomial expression for the inverse matrix function of the lowest possible degree. Although the minimal polynomial is important to know for a variety of reasons, it is usually difficult to compute even for matrices of small dimension (e.g., see [2, p. 186-188]).

For some special classes of matrices in $M_n(\mathbb{C})$, the minimal polynomial can be obtained rather easily. One such class is that of idempotent matrices which are a non-zero, non-identity matrix that has the property that $A^2 = A$. Idempotent matrices have the minimal polynomial of $q_A(t) = t(t-1)$. More generally, any non-zero, non-identity matrix such that $A^n = A$, where n is the minimal degree that satisfies that property, has the minimal polynomial $q_A(t) = t(t-1)(t^{n-2} + \cdots + t^2 + t + 1)$. The main theme of this dissertation is to devise a more practical algorithm that computes the minimal polynomial for any matrix in $M_n(\mathbb{C})$.

There are a few known methods for computing the minimal polynomial of a matrix $A \in M_n(\mathbb{C})$. One method relies on knowing the characteristic polynomial of a given matrix. Let $\lambda_i \in \mathbb{C}, i = 1, \ldots, k$ be the distinct eigenvalues of a matrix $A \in M_n(\mathbb{C})$ such that the characteristic polynomial of matrix A is known and has been factored as $p_A(t) = (t - \lambda_1)^{n_1} \ldots (t - \lambda_k)^{n_k}$, where $n_1 + \cdots + n_k = n$, $k \leq n$. Then the minimal polynomial is determined by computing combinations of $(A - \lambda_1 I)^{q_1} \ldots (A - \lambda_k I)^{q_k}$ for each $q_i \leq n_i$ with $i = 1, \ldots, k$ until the first occurence when $q_A(A) = (A - \lambda_1 I)^{q_1} \ldots (A - \lambda_k I)^{q_k} = 0$. This method is not a very effective way of obtaining the minimal polynomial. Note that this method requires knowledge of all of the distinct eigenvalues of the matrix, and knowing the eigenvalues before having the minimal polynomial is a very special case.

There are other methods of computing the minimal polynomial that do not rely on knowing the characteristic polynomial of a given $A \in M_n(\mathbb{C})$. One such method

is outlined in [7, page 148]. This method transforms matrix A into a vector in $\mathbb{C}^{n^2}$ using the isomorphism defined by
$T(A) = vec(A) \equiv (a_{11}, \ldots, a_{n1}, a_{12}, \ldots, a_{n2}, \ldots, a_{1n}, \ldots, a_{nn})^T$. Then the Gram-Schmidt process is applied to the vectors $\{vec(I), vec(A), \ldots, vec(A^i)\}$, where $vec(A^i)$ gives the first occurence of the linear dependency during the Gram Schmidt's process. It is at this point where the coefficients of the minimal polynomial can be obtained since $vec(A^i)$ can be written as a linear combination of the previous vectors. The coefficients of the minimal polynomial are the Fourier coefficients for $vec(A^i)$ with respect to the orthonormal basis obtained from $\{vec(I), vec(A), \ldots, vec(A^{i-1})\}$ by the Gram-Shcmidt's process. A benifit of this method is that it computes the minimal polynomial of any matrix. We observe, however, that the method is not practical to use for matrices of large size, but we will use the idea of isomorphically transforming a matrix to a vector in $\mathbb{C}^{n^2}$ in our algorithm.

A paper by S. Bialas and M. Bialas [1] also uses the aforementioned isomorphic transformation. In their method the authors map the matrix isomorphically to vectors in $\mathbb{C}^{n^2}$ using the *vec* operation, but they do not use the Gram-Schmidt process. Their method begins by computing powers of the matrix A to create the set, $\{I, A, A^2, \ldots, A^n\}$. Then they convert the matrices to column vectors by applying the *vec* operation. Using the set $\{vec(I), vec(A), \ldots, vec(A^n)\}$, the method constructs a huge $n^2 \times (n + 1)$ matrix and uses Gaussian elimination to obtain the minimal polynomial. Clearly, the method is extremely inefficient since all the powers of the matrix have to be computed before the Gaussian elimination process begins. This method is especially impractical to use for computing the minimal polynomial of a large sized matrix.

CHAPTER 2

How to compute the minimal polynomial in a computationally simple way

This chapter develops an algorithm that computes the minimal polynomial of any given matrix $A \in M_n(\mathbb{C})$. From here on we will refer to this algorithm as the MP algorithm. Before describing the MP algorithm, we introduce the concepts that form the theoretical basis of our work, after which we introduce the MP algorithm. Once the MP algorithm has been established, we present a modified version of the MP algorithm that is easier to apply. We call the modified MP algorithm the MMP algorithm. Throughout the chapter, examples are provided to illustrate how the algorithms are implemented.

2.1 The MP algorithm

Our algorithm depends on the set $D' = \{A^0, A, A^2, \ldots, A^n\}$ being linearly dependent. We find a linearly dependent set with the least number of elements in it by computing $A, A^2, \ldots, A^k$ successively, for each $k \leq n$. For this reason we make the following definition.

Definition 2.1 *Let $S = \{v_0, v_1, \ldots, v_n\} \subseteq \mathbb{C}_{n^2}$ be an ordered set. We say the ordered set of vectors $\{v_0, v_1, \ldots, v_k\}$ with $k \leq n$ is a* ***minimal linearly dependent*** *set, if the set $\{v_0, v_1, \ldots, v_k\}$ is linearly dependent such that $\{v_0, v_1, \ldots, v_j\}$ is linearly independent for all $j < k$.*

Another way to say that the ordered set $\{v_0, v_1, \ldots, v_k\}$ with $k \leq n$ is minimal linearly dependent is that there exists scalars, $\alpha_0, \ldots, \alpha_k$, which are not all zero such that $\alpha_0 v_0 + \cdots + \alpha_k v_k = 0$ and if $\beta_0 v_0 + \cdots + \beta_j v_j = 0$, then $\beta_0 = \cdots = \beta_j = 0$ for $0 \leq j < k$. The nontrivial zero linear combination $\alpha_0 v_0 + \cdots + \alpha_k v_k = 0$ is said to be the minimal zero linear combination. Observe that $\alpha_k \neq 0$ in any minimal zero linear combination $\alpha_0 v_0 + \cdots + \alpha_{k-1} v_{k-1} + \alpha_k v_k = 0$. Thus, we may assume $\alpha_k = 1$ for a minimal zero linear combination, and say it is monic.

Proposition 2.1 *A monic minimal zero linear combination for a minimal linearly dependent set is unique.*

Proof. Let $\{v_0, \ldots, v_k\}$ be a minimal linearly dependent set. Suppose the monic minimal zero linear combination is not unique. Then there exists scalars $\alpha_i, i = 0, \ldots, k-1$, not all zero, and scalars $\beta_i, i = 0, \ldots, k-1$, not all zero, such that $\alpha_0 v_0 + \cdots + \alpha_{k-1} v_{k-1} + v_k = 0$ and $\beta_0 v_0 + \cdots + \beta_{k-1} v_{k-1} + v_k = 0$. Subtracting the two minimal linear combinations, we obtain $(\alpha_0 - \beta_0) v_0 + \cdots + (\alpha_{k-1} - \beta_{k-1}) v_{k-1} + 0 v_k = 0$. If $\alpha_i - \beta_i \neq 0$ for any $i = 1, \ldots, k-1$, then $\{v_0, \ldots, v_{k-1}\}$ is a linearly dependent set which contradicts the set $\{v_0, \ldots, v_k\}$ being a minimal linearly dependent set. Thus, $\alpha_i = \beta_i$ for $i = 0, \ldots, k-1$, and we conclude that a monic minimal zero linear combination of a minimal linearly dependent set is unique. $\square$

The following definition is an important operation on matrices used in the MP algorithm.

Definition 2.2 *Let $A = [a_{ij}] \in M_n(\mathbb{C})$. Then a row vector, denoted by **rvec(A)**, is defined by $rvec(A) = [a_{11}, \ldots, a_{1n} : a_{21}, \ldots, a_{2n} : \ldots : a_{n1}, \ldots, a_{nn}] \in \mathbb{C}_{n^2}$.*

The $rvec(A)$ operation on $A \in M_n(\mathbb{C})$ places the rows of matrix A consecutively to create a row vector. It should be noted that $rvec(A)$ is a simple modification of the usual $vec(A) = [a_{11}, \ldots, a_{n1} : a_{12}, \ldots, a_{n2} : \ldots : a_{1n}, \ldots, a_{nn}]^T \in \mathbb{C}^{n^2}$ given in matrix theory books. Note that $rvec(A) = (vec(A^T))^T$. We record this as a proposition.

Proposition 2.2 *Let $A \in M_n(\mathbb{C})$. Then $rvec(A) = (vec(A^T))^T$.*

Proof. Let $A = \begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix} \in M_n(\mathbb{C})$ where $r_i \in M_{1\times n}(\mathbb{C})$, $i = 1, \ldots, n$, are the rows of A. Then $rvec(A) = [r_1, \ldots, r_n] = \begin{bmatrix} r_1^T \\ \vdots \\ r_n^T \end{bmatrix}^T = (vec(r_1^T, \ldots, r_n^T))^T = vec(A^T)^T$. $\square$

To compute the minimal polynomial, it will be useful to view the problem in a different way. Our method for computing the minimal polynomial is to search for the minimal linearly dependent set in $\mathbb{C}_{n^2}$ instead of looking for the minimal linearly dependent set in $M_n(\mathbb{C})$. For this purpose, we define the map $f : M_n(\mathbb{C}) \to \mathbb{G} = \mathbb{C}_{n^2}$ by $f(A) \equiv rvec(A)$, and show f to be an isomorphism in the next proposition. The motivation behind transforming a matrix isomorphically to $\mathbb{C}_{n^2}$ is that it is easier to find the minimal linearly dependent set in $\mathbb{C}_{n^2}$. Once the minimal linearly dependent set is obtained, we find the coefficients of the minimal polynomial for $A \in M_n(\mathbb{C})$. This idea appears in [7, p.148]. At this point we would like to mention that there is no need to restrict ourselves to a specific isomorphism such as the one defined as *rvec*. We may choose an isomorphism that is most conveinvient for the class of matrices being considered. However, a natural choice for transforming $A \in M_n(\mathbf{C})$ is through the use of the isomorphism *rvec*.

The following proposition shows that $f(A) = rvec(A)$ is in fact an isomorphism.

Proposition 2.3 *The map $f : M_n(\mathbb{C}) \to \mathbb{C}_{n^2}$ be defined by $f(A) = rvec(A)$ is an isomorphism.*

Proof. Let $f : M_n(\mathbb{C}) \to \mathbb{C}_{n^2}$ be defined by $f(A) = rvec(A)$. Clearly, $f(kA + B) = rvec(kA+B) = k \cdot rvec(A) + rvec(B) = kf(A) + f(B)$. So f is linear. Let $A = \begin{bmatrix} A_1 \\ \vdots \\ A_n \end{bmatrix}$ and $B = \begin{bmatrix} B_1 \\ \vdots \\ B_n \end{bmatrix} \in M_n(\mathbb{C})$, where $A_i, B_i, i = 1, \ldots, n$ are the rows of the matrices A and B, respectively. Then $f(A) = f(B)$ implies that $rvec(A) = rvec(B)$, or $\begin{bmatrix} A_1, & \ldots, & A_n \end{bmatrix} = \begin{bmatrix} B_1, & \ldots, & B_n \end{bmatrix}$. Thus, $A_i = B_i$ for each $i = 1, \ldots, n$. Hence $A = B$, and f is one-to-one. For any $\begin{bmatrix} C_1, & \ldots, & C_n \end{bmatrix} \in \mathbb{C}_{n^2}$ where $C_i \in \mathbb{C}_n$, let $A = \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix}$. Then $f(A) = rvec(A) = [C_1, \ldots, C_n]$. Thus f is onto. Together these show that f is an isomorphism, and the proof is complete. $\square$

We now show that $\{rvec(I), rvec(A), \ldots, rvec(A^n)\}$ is a linearly dependent set.

Proposition 2.4 *Let $A \in M_n(\mathbb{C})$ and $f(A) = rvec(A)$. Then the set*

$$S = \{v_0 = f(I), v_1 = f(A), v_2 = f(A^2), \ldots, v_n = f(A^n)\}$$

is linearly dependent.

Proof. Suppose $S = \{rvec(I), \ldots, rvec(A^n)\}$. The Cayley-Hamilton theorem asserts there is a linear combination of the matrices $\{I, \ldots, A^n\}$ such that $A^n + \alpha_{n-1}A^{n-1} + \cdots + \alpha_1 A + \alpha_0 I = 0$. Then, since f is an isomorphism, we have $0 = f(0) = f(A^n + \alpha_{n-1}A^{n-1} + \cdots + \alpha_1 A + \alpha_0 I) = f(A^n) + \alpha_{n-1}f(A^{n-1}) + \cdots + \alpha_1 f(A) + \alpha_0 f(I) = v_n + \alpha_{n-1}v_{n-1} + \cdots + \alpha_1 v_1 + \alpha_0 v_0$. Thus the set S is a linearly dependent set. $\square$

The previous two propositions show that the coefficients of the minimal polynomial can be obtained from the minimal linear combination of the row vectors $rvec(I), rvec(A), \ldots, rvec(A^k)$ for $k \leq n$. We now proceed to find this minimal linear combination for $A \in M_n(\mathbb{C})$.

In $\mathbb{C}_{n^2}$ we use Gaussian elimination as an easy way to determine whether vectors are linearly dependent in order to find the minimal linearly dependent set. We begin by placing $rvec(I_n)$ into a matrix with $rvec(A)$, and use Gaussian elimination to see if they are linearly dependent. If they are linearly dependent, then we have found the minimal linearly dependent set. If they are not linearly dependent, then we place $rvec(I_n)$ and $rvec(A)$ into a matrix with $rvec(A^2)$. We use Gaussian elimination to check their linear dependence. When we successively place vectors into a matrix to determine if they are linearly dependent, we will only use the last row of the matrix in the Gaussian elimination. We continue this process on $rvec(I_n)$, $rvec(A)$, $\ldots$, $rvec(A^k)$ where $k \le n$ until the first instance when the vectors are linearly dependent. Once they have been shown to be linearly dependent, we have found the minimal linearly dependent set.

Let $v_i = rvec(A^i)$ for $i = 0, \ldots, n$. By Proposition (2.4) we know that $\{v_0, \ldots, v_n\}$ contains the minimal linearly dependent set. One advantage of our method over existing methods is that it is not necessary to calculate all the v_i's in order to obtain the minimal polynomial.

We define the following augmented matrix which plays a significant role in the MP algorithm.

Definition 2.3 *Let $\{v_0, \ldots, v_k\} \subseteq \mathbb{C}_{n^2}$ and $B_{k+1} \in M_{k+1}(\mathbb{C})$ be given. We define $G_{\{v_0,\ldots,v_k\}}(B_{k+1}) \in M_{k+1,n^2+k+1}(\mathbb{C})$ to be the matrix,*

$$G_{\{v_0,\ldots,v_k\}}(B_{k+1}) \equiv \left[\begin{array}{c|c} v_0 & \\ v_1 & \\ \vdots & B_{k+1} \\ v_k & \end{array}\right].$$

The matrix $G_{\{v_0,\ldots,v_k\}}(B_{k+1})$ will be called the ***Gaussian updating (GU) matrix.***

Before stating the MP algorithm, we first discribe the role played by the Gaussian updating matrix. We begin with the Gaussian updating matrix $G_{\{v_0\}}(B_1) =$

$[v_0 \| 1] = [e_1^T \quad e_2^T \quad \cdots \quad e_n^T \quad \| \quad 1]$ where $B_1 = I_1$. Next, we create the GU matrix $G_{\{v_0,v_1\}}(B_2)$ where $B_2 = \begin{bmatrix} B_1 & 0 \\ 0 & 1 \end{bmatrix} \in M_2(\mathbb{C})$. Gaussian row operations are performed to determine whether or not v_0 and v_1 are linearly dependent. Out of this we have a new GU matrix, $G_{\{v_0,v_1'\}}(B_2')$, where v_1' is the vector obtained from v_1 in the Gaussian elimination, and B_2' is the matrix obtained from B_2 by the Gaussian elimination. Successively, we construct the GU matrix $G_{\{v_0,v_1',\ldots,v_{k-1}',v_k\}}(B_{k+1})$, where $B_{k+1} = \begin{bmatrix} B_k' & 0 \\ 0 & e_k^T \end{bmatrix}$. Gaussian elimination is used to determine whether or not the newly introduced vector, v_k, is linearly dependent to the vectors in the set $\{v_0, v_1', \ldots, v_{k-1}'\}$. From this we obtain a new GU matrix, $G_{\{v_0,v_1',\ldots,v_{k-1}',v_k'\}}(B_{k+1}')$, where v_k' is the vector obtained from v_k in the Gaussian elimination, and B_{k+1}' is the matrix obtained from B_{k+1} in the Gaussian elimination. We know that this process must produce a zero vector, v_k', for some $k \leq n$ by Proposition (2.4), and finding this zero vector completes the updating process.

We now describe how to construct the matrix B_{k+1} used in the GU matrix. Let $B_1 = I_1$. Create $B_2 = \begin{bmatrix} B_1 & 0 \\ 0 & e_1^T \end{bmatrix}$ and construct the GU matrix $G_{\{v_0,v_1\}}(B_2)$. We use Gaussian elimination to obtain the GU matrix $G_{\{v_0,v_1'\}}(B_2')$ and pay special attention to the columns of B_2'. Since the matrix B_2' is obtained from B_2 through Gaussian elimination we see that the first column of B_2' is the coefficient of I_n in the linear combination of v_1' and the second column of B_2' is the coefficient of A in the linear combination of v_1'. We can generalize this and consider when $G_{\{v_0,v_1',\ldots,v_k'\}}(B_{k+1}')$ is obtained from $G_{\{v_0,v_1'\ldots,v_{k-1}',v_k\}}(B_{k+1})$ using Gaussian elimination. Then the first column of B_{k+1}' is the coefficient of I_n in the linear combination of v_k'; the second column of the matrix B_{k+1}' is the coefficient of A in the linear combination of v_k'; the third column of the matrix B_{k+1}' is the coefficient of A^2 in the linear combination of v_k'; and, in general the k-th column of the matrix B_{k+1}' is the coefficient of A^k in the linear combination of v'_k. The vector v'_k is the *rvec* of the linear combination of the matrices $I_n, A, A^2, \ldots, A^k$, the coefficients of the minimal polynomial are given in the last row of B'_{k+1}.

As stated earlier, it is not necessary to compute all the powers of A in order to compute each v_i. Since the set $D' = \{A^0, A, A^2, \ldots, A^n\}$ contains a linearly dependent set of vectors with $k \leq n$, we only need to compute the first k powers of the matrix A. We now give our algorithm to compute the minimal polynomial of any given matrix $A \in M_n(\mathbb{C})$.

The Minimal Polynomial Algorithm (MP)

For a given $A \in M_n(\mathbb{C})$, let $v_i = rvec(A^i)$, and do the following.

Step 1. (Initialization). Create $G_{\{v_0\}}(I_1)$, set $v_0 \equiv v'_0$, $i = 1$, and $B_1 \equiv I_1$.

Step 2. Compute v_i and construct $G_{\{v'_0,\ldots,v'_{i-1},v_i\}}(B_{i+1})$ where $B_{i+1} \equiv \begin{bmatrix} B'_i & 0 \\ 0 & e_i^T \end{bmatrix}$. Use Gaussian elimination to obtain $G_{\{v'_0,\ldots,v'_{i-1},v'_i\}}(B'_{i+1})$.

- If $v'_i \equiv 0$ stop and proceed to Step 3.
- If $v'_i \neq 0$, increment i by 1 and repeat Step 2.

Step 3. For $i = k$ such that $v'_k \equiv 0$, the entries of the last row of B'_{k+1}, $b_{k+1,j} \in \mathbb{C}$ with $j = 1, \ldots, k+1$, are the coefficients of the minimal polynomial of the matrix $A \in M_n(\mathbb{C})$.

Here is an example that illustrates our algorithm.

Example 2.1 In this example we compute the minimal polynomial for the matrix

$$A = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 2 & 1 \\ 2 & 0 & 1 \end{bmatrix} \in M_3(\mathbb{C}).$$

Starting with the vector $v_0 = v_0' = rvec(I_3) = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 \end{bmatrix}$, we construct the initial GU matrix

$$G_{\{v_0'\}}(B_1) = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 \end{bmatrix}.$$

Next we compute

$$v_1 = rvec(A) = \begin{bmatrix} 1 & 1 & 0 & : & -1 & 2 & 1 & : & 2 & 0 & 1 \end{bmatrix}.$$

Now we can construct the GU matrix

$$G_{\{v_0',v_1\}}(B_2) = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 \\ 1 & 1 & 0 & : & -1 & 2 & 1 & : & 2 & 0 & 1 & \| & 0 & 1 \end{bmatrix}.$$

Using Gaussian row elimination, we check whether v_0' and v_1 are linearly dependent. Applying the elementary operation $R_2 - R_1 \to R_2$, the resulting GU matrix is

$$G_{\{v_0',v_1'\}}(B_2') = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 \end{bmatrix}.$$

Since $v_1' \neq 0$, we know v_0' and v_1' are linearly independent and the algorithm continues by computing

$$v_2 = rvec(A^2) = \begin{bmatrix} 0 & 3 & 1 & : & -1 & 3 & 3 & : & 4 & 2 & 1 \end{bmatrix}$$

and creating the GU matrix

$$G_{\{v_0',v_1',v_2\}}(B_3) = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 & 0 \\ 0 & 3 & 1 & : & -1 & 3 & 3 & : & 4 & 2 & 1 & \| & 0 & 0 & 1 \end{bmatrix}.$$

We check to see whether v_0', v_1' and v_2 are linearly dependent using the elementary row operations $R_3 - 3R_2 \to R_3$. This gives us the GU matrix

$$G_{\{v_0',v_1',v_2'\}}(B_3') = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 & 0 \\ 0 & 0 & 1 & : & 2 & 0 & 0 & : & -2 & 2 & 1 & \| & 3 & -3 & 1 \end{bmatrix}.$$

Again, since $v_2' \neq 0$, we know that v_0', v_1', and v_2 are not linearly dependent. Therefore we continue the algorithm by computing

$$v_3 = rvec(A^3) = \begin{bmatrix} -1 & 6 & 4 & : & 2 & 5 & 6 & : & 4 & 8 & 3 \end{bmatrix}$$

and constructing the GU matrix

$$G_{\{v_0',v_1',v_2',v_3\}}(B_4) = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & : & 2 & 0 & 0 & : & -2 & 2 & 1 & \| & 3 & -3 & 1 & 0 \\ -1 & 6 & 4 & : & 2 & 5 & 6 & : & 4 & 8 & 3 & \| & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Following the MP algorithm, we check whether the vectors v_0', v_1', v_2' and v_3 are linearly dependent using the elementary row operations $R_4 + R_1 \to R_4, R_4 - 6R_2 \to R_4$, and $R_4 - 4R_3 \to R_4$. This results in the GU matrix

$$G_{\{v_0',v_1',v_2',v_3'\}}(B_4') = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & : & 2 & 0 & 0 & : & -2 & 2 & 1 & \| & 3 & -3 & 1 & 0 \\ 0 & 0 & 0 & : & 0 & 0 & 0 & : & 0 & 0 & 0 & \| & -5 & 6 & -4 & 1 \end{bmatrix}.$$

Since $v'_3 \equiv 0$, the algorithm terminates and the coefficients of the minimal polynomial can be read off from the last row of the matrix

$$B'_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 3 & -3 & 1 & 0 \\ -5 & 6 & -4 & 1 \end{bmatrix}.$$

The minimal polynomial for the matrix A is $q_A(t) = -5 + 6t - 4t^2 + t^3$.

2.2 The modified minimal polynomial (MMP) algorithm

Example (2.1) from the previous section allows us to make a small observation that will improve our algorithm. In the algorithm, v_k should be obtained from v'_{k-1} instead of v_{k-1}. This will reduce the number of computations since v'_{k-1} will have more zeros than v_k, which are produced during the Gaussian elimination. In order to accomplish this, we introduce the Kronecker product.

Definition 2.4 *[8, Definition 4.2.1] The* ***Kronecker product*** *of* $A = [a_{ij}] \in M_{m,n}(\mathbb{C})$ *and* $B = [b_{ij}] \in M_{p,q}(\mathbb{C})$ *is denoted by* $A \otimes B$ *and is defined to be a matrix of the following form,*

$$A \otimes B \equiv \begin{bmatrix} a_{1,1}B & \cdots & a_{1,n}B \\ \vdots & \ddots & \vdots \\ a_{m,1}B & \cdots & a_{m,n}B \end{bmatrix} \in M_{mp,nq}(\mathbb{C}).$$

Proposition 2.5 *[8, Theorem 4.3.1] Let* A, B *and* $C \in M_n(\mathbb{C})$*, then* $vec(ABC) = (C^T \otimes A)vec(B)$.

Proposition 2.6 *Let A and B be matrices so that the product BA is defined. Then $rvec(BA) = rvec(B)(I \otimes A)$.*

Proof. We know that $rvec(A)$ and $vec(A)$ are related by $rvec(A) = (vec(A^T))^T$ as seen in Proposition (2.2). We apply this fact and Proposition (2.5) to calculate $rvec(BA) = vec((BA)^T)^T = vec(A^T B^T)^T = ((I \otimes A^T)vec(B^T))^T = ((I \otimes A)^T vec(B^T))^T = vec(B^T)^T(I \otimes A) = rvec(B)(I \otimes A)$. $\square$

Proposition (2.6) gives us the tool we need to write the v_k in terms of the row vector v'_{k-1}. To see how, note that v'_{k-1} is the linear combination of $\{v_0, \ldots, v_{k-2}, v_{k-1}\}$, say $v'_{k-1} = \sum_{i=0}^{k-1} b_i v_i = \sum_{i=0}^{k-1} b_i rvec(A^i) = rvec(\sum_{i=0}^{k-1} b_i A^i)$. Notice that the linear combination $\sum_{i=0}^{k-1} b_i A^i$ is just some matrix. So we apply Proposition (2.6) to obtain the recurrence relation $v'_{k-1}(I \otimes A) = rvec(\sum_{i=0}^{k-1} b_i A^i)(I \otimes A) = rvec(\sum_{i=0}^{k-1} b_i A^i A) = v_k$. We note that this new v_k is different from the $rvec$ of A^k in the MP algorithm. Rather, this new v_k is the $rvec$ of the linear combination of $\sum_{i=0}^{k-1} b_i A^i A$.

Once we have computed v_k for some $k \leq n$ by using Proposition (2.6), we need the $(k+1)$-th row of the lower triangular matrix B_{k+1} in the GU matrix. In light of Proposition (2.6), we see that the relationship $v_k = v'_{k-1}(I \otimes A)$ effectively multiplies the $rvec$ of the linear combination of the k-th row in the GU matrix by A. Thus, the $(k+1)$-th row of the matrix B_{k+1} may be obtained by shifting the entries to the right 1 entry. To demonstrate this suppose that $[b_1 \ldots b_k \ 0]$ is the k-th row of the matrix B'_k then $[0 \ b_1 \ldots b_k]$ is the last row in the new matrix B_{k+1}. A visual representation of this is helpful. Suppose we have computed the following GU matrix

$$G_{\{v'_0,\ldots,v'_{k-1}\}}(B'_k) = \left[\begin{array}{c||ccc} v'_0 & & & \\ v'_1 & & B'_{k-1} & \\ \vdots & & & \\ v'_{k-1} & b_1 & \ldots & b_k \end{array}\right].$$

Then v'_{k-1} is the *rvec* of the linear combination of $I, \ldots, A^{k-1}$. In other words $v'_{k-1} = b_0 I + \cdots + b_{k-1} A^{k-1}$. The updated GU matrix after adding the vector v_k is

$$G_{\{v'_0,\ldots,v'_{k-1},v_k\}}(B_{k+1}) = \begin{bmatrix} v'_0 & \| & & & \\ v'_1 & \| & B'_{k-1} & & \\ \vdots & \| & & & \\ v'_{k-1} & \|b_1 & \ldots & b_k & 0 \\ v_k & \|0 & b_1 & \ldots & b_k \end{bmatrix}.$$

Shifting each element of the k-th row of the matrix B'_k to the right one entry ($[0\ b_1 \ldots b_k]$) gives us the $(k+1)$-th row of the new matrix B_{k+1}, which corresponds with the coefficients of the linear combination $v_{k+1} = rvec(b_1 A + \cdots + b_k A^{k+1})$. We now give a modified version of the MP algorithm that will compute the minimal polynomial of any matrix $A \in M_n(\mathbb{C})$ and the modified algorithm is referred to as the modified minimal polynomial algorithm (MMP).

The Modified Minimal Polynomial Algorithm (MMP)

For a given $A \in M_n(\mathbb{C})$, do the following.

Step 1. Create $G_{\{v_0\}}(I_1)$, set $v_0 = rvec(I_n) \equiv v_0'$, set $i = 1$, and $B_1 \equiv I_1$

Step 2. Compute $v_i = v_{i-1}'(I \otimes A)$ and construct the GU matrix $G_{\left\{v_0',\ldots,v_{i-1}',v_i\right\}}(B_{i+1})$ where $B_{i+1} \equiv \begin{bmatrix} B_i' & 0 \\ 0 & b \end{bmatrix}$, such that $b \in \mathbb{C}_i$ are the entries of the last row of B_i'. Use Gaussian elimination to obtain $G_{\left\{v_0',\ldots,v_{i-1}',v_i'\right\}}(B_{i+1}')$.

- If $v_i' \equiv 0$ stop and proceed to Step 3.
- If $v_i' \neq 0$, increment i by 1 and repeat Step 2.

Step 3. For $i = k$ such that $v_k' \equiv 0$, the entries of the last row of B_{k+1}', $b_{k+1,j} \in \mathbb{C}$ for $j = 1, \ldots, k+1$, are the coefficients of the minimal polynomial of the matrix $A \in M_n(\mathbb{C})$.

In the following example we compute the minimal polynomial of the same matrix used in Example (2.1), but here we use the MMP algorithm to show how it is more efficient than the MP algorithm. The efficiency comes from less computations each time Gaussian ellimination. The savings will be significant if the modified algorithm is applied to a certain sparse matrix.

Example 2.2 In this example we illustrate our modified algorithm for calculating the minimal polynomial of the matrix

$$A = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 2 & 1 \\ 2 & 0 & 1 \end{bmatrix} \in M_3(\mathbb{C}).$$

Starting with the vector $v_0 = v_0' = rvec(I_3) = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 \end{bmatrix}$, we construct the initial GU matrix

$$G_{\{v_0'\}}(B_1) = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 \end{bmatrix}.$$

Next we compute

$$\begin{aligned} v_1 &= v_0'(I \otimes A) \\ &= \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 0 & : & -1 & 2 & 1 & : & 2 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Now we can construct the GU matrix

$$G_{\{v_0',v_1\}}(B_2) = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 \\ 1 & 1 & 0 & : & -1 & 2 & 1 & : & 2 & 0 & 1 & \| & 0 & 1 \end{bmatrix}.$$

Using Gaussian row elimination, we check whether v_0' and v_1 are linearly dependent. Applying the elementary operation $R_2 - R_1 \to R_2$, the resulting GU matrix is

$$G_{\{v_0',v_1'\}}(B_2') = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 \end{bmatrix}.$$

Since $v_1' \neq 0$, we know v_0' and v_1 are linearly independent and the algorithm continues by computing

$$\begin{aligned} v_2 &= v_1'(I \otimes A) \\ &= \begin{bmatrix} 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{bmatrix} \\ &= \begin{bmatrix} -1 & 2 & 1 & : & 0 & 1 & 2 & : & 2 & 2 & 0 \end{bmatrix} \end{aligned}$$

and creating the GU matrix

$$G_{\{v_0',v_1',v_2\}}(B_3) = \left[\begin{array}{ccccccccccccccc} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 & 0 \\ -1 & 2 & 1 & : & 0 & 1 & 2 & : & 2 & 2 & 0 & \| & 0 & -1 & 1 \end{array}\right].$$

Using the elementary row operations $R_3 + R_1 \to R_3$ and $R_3 - 2R_2 \to R_3$ we obtain the GU matrix

$$G_{\{v_0',v_1',v_2'\}}(B_3') = \left[\begin{array}{ccccccccccccccc} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 & 0 \\ 0 & 0 & 1 & : & 2 & 0 & 0 & : & -2 & 2 & 1 & \| & 3 & -3 & 1 \end{array}\right].$$

Again, since $v_2' \neq 0$, we proceed to the next step by computing

$$\begin{aligned} v_3 &= v_2'(I \otimes A) \\ &= \left[\begin{array}{ccccccccccc} 0 & 0 & 1 & : & 2 & 0 & 0 & : & -2 & 2 & 1 \end{array}\right] \left[\begin{array}{ccc} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{array}\right] \\ &= \left[\begin{array}{ccccccccccc} 2 & 0 & 1 & : & 2 & 2 & 0 & : & -2 & 2 & 3 \end{array}\right]. \end{aligned}$$

and creating the GU matrix

$$G_{\{v_0',v_1',v_2',v_3\}}(B_4) = \left[\begin{array}{cccccccccccccccc} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & : & 2 & 0 & 0 & : & -2 & 2 & 1 & \| & 3 & -3 & 1 & 0 \\ 2 & 0 & 1 & : & 2 & 2 & 0 & : & -2 & 2 & 3 & \| & 0 & 3 & -3 & 1 \end{array}\right].$$

Using the elementary row operations $R_4 - 2R_1 \to R_4$ and $R_4 - R_3 \to R_4$, the resulting GU matrix is

$$G_{\{v_0',v_1',v_2',v_3'\}}(B_4') = \left[\begin{array}{cccccccccccccccc} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & : & 2 & 0 & 0 & : & -2 & 2 & 1 & \| & 3 & -3 & 1 & 0 \\ 0 & 0 & 0 & : & 0 & 0 & 0 & : & 0 & 0 & 0 & \| & -5 & 6 & -4 & 1 \end{array}\right].$$

Since $v_3' \equiv 0$, the algorithm terminates and the coefficients of the minimal polynomial can be read off from the last row of the matrix

$$B_4' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 3 & -3 & 1 & 0 \\ -5 & 6 & -4 & 1 \end{bmatrix}.$$

The minimal polynomial for the matrix A is $q_A(t) = -5 + 6t - 4t^2 + t^3$.

The next example is included to show the computational savings of the MMP algorithm when compared to the algorithm given in [1] that requires computing $A, A^2, \ldots, A^n$ for $A \in M_n(\mathbb{C})$.

Example 2.3 We use the MMP algorithm to calculate the minimal polynomial for the matrix

$$A = \begin{bmatrix} 3 & -1 & -1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{bmatrix} \in M_4(\mathbb{C}).$$

We start with $v_0' = \begin{bmatrix} 1 & 0 & 0 & 0 & : 0 & 1 & 0 & 0 & : & 0 & 0 & 1 & 0 : & 0 & 0 & 0 & 1 \end{bmatrix}$ and create the GU matrix

$$G_{\{v_0'\}}(B_1) = \begin{bmatrix} 1 & 0 & 0 & 0 & : & 1 & 0 & 0 & 0 & : & 0 & 0 & 1 & 0 & : & 0 & 0 & 0 & 1 & \| & 1 \end{bmatrix}.$$

Next we compute

$$\begin{aligned} v_1 &= v_0'(I \otimes A) \\ &= \begin{bmatrix} e_1^T & e_2^T & e_3^T & e_4^T \end{bmatrix} \begin{bmatrix} A & 0 & 0 & 0 \\ 0 & A & 0 & 0 \\ 0 & 0 & A & 0 \\ 0 & 0 & 0 & A \end{bmatrix} \\ &= \begin{bmatrix} 3 & -1 & -1 & 0 & : 1 & 1 & -1 & 0 & : & 1 & -1 & 1 & 0 & : 1 & -1 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Now we can construct the GU matrix

$$G_{\{v_0',v_1\}}(B_2) =$$

$$\left[\begin{array}{cccccccccccccccccccccc} 1 & 0 & 0 & 0 & : & 0 & 1 & 0 & 0 & : & 0 & 0 & 1 & 0 & : & 0 & 0 & 0 & 1 & \| & 1 & 0 \\ 3 & -1 & -1 & 0 & : & 1 & 1 & -1 & 0 & : & 1 & -1 & 1 & 0 & : & 1 & -1 & 0 & 1 & \| & 0 & 1 \end{array}\right].$$

Using the elementary row operation $R_2 - 3R_1 \to R_2$ we obtain the GU matrix

$$G_{\{v_0',v_1'\}}(B_2') =$$

$$\left[\begin{array}{cccccccccccccccccccccc} 1 & 0 & 0 & 0 & : & 0 & 1 & 0 & 0 & : & 0 & 0 & 1 & 0 & : & 0 & 0 & 0 & 1 & \| & 1 & 0 \\ 0 & -1 & -1 & 0 & : & 1 & -2 & -1 & 0 & : & 1 & -1 & -2 & 0 & : & 1 & -1 & 0 & -2 & \| & -3 & 1 \end{array}\right].$$

Since $v_1' \neq 0$, we continue the algorithm by computing

$$\begin{aligned} v_2 &= v_1'(I \otimes A) \\ &= \left[\begin{array}{ccccccccccccccccccc} 0 & -1 & -1 & 0 & : & 1 & -2 & -1 & 0 & : & 1 & -1 & -2 & 0 & : & 1 & -1 & 0 & -2 \end{array}\right] \begin{bmatrix} A & 0 & 0 & 0 \\ 0 & A & 0 & 0 \\ 0 & 0 & A & 0 \\ 0 & 0 & 0 & A \end{bmatrix} \\ &= \left[\begin{array}{cccc} -2e_1^T & -2e_2^T & -2e_3^T & -2e_4^T \end{array}\right]. \end{aligned}$$

and constructing the GU matrix

$$G_{\{v_0',v_1',v_2\}}(B_3) = \left[\begin{array}{ccccccccccccccccccccccc} 1 & 0 & 0 & 0 & : & 0 & 1 & 0 & 0 & : & 0 & 0 & 1 & 0 & : & 0 & 0 & 0 & 1 & \| & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 & : & 1 & -2 & -1 & 0 & : & 1 & -1 & -2 & 0 & : & 1 & -1 & 0 & -2 & \| & -3 & 1 & 0 \\ -2 & 0 & 0 & 0 & : & 0 & -2 & 0 & 0 & : & 0 & 0 & -2 & 0 & : & 0 & 0 & 0 & -2 & \| & 0 & -3 & 1 \end{array}\right].$$

Using the elementary row operation $R_3 + 2R_1 \to R_3$ we obtain the GU matrix

$$G_{\{v_0',v_1',v_2'\}}(B_3') = \left[\begin{array}{ccccccccccccccccccccccc} 1 & 0 & 0 & 0 & : & 0 & 1 & 0 & 0 & : & 0 & 0 & 1 & 0 & : & 0 & 0 & 0 & 1 & \| & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 & : & 1 & -2 & -1 & 0 & : & 1 & -1 & -2 & 0 & : & 1 & -1 & 0 & -2 & \| & -3 & 1 & 0 \\ 0 & 0 & 0 & 0 & : & 0 & 0 & 0 & 0 & : & 0 & 0 & 0 & 0 & : & 0 & 0 & 0 & 0 & \| & 2 & -3 & 1 \end{array}\right]$$

Since $v'_2 \equiv 0$, the algorithm terminates, and the coefficients of the minimal polynomial can be read off from the last row of the matrix B'_3. The minimal polynomial of the matrix A is $q_A(t) = 2 - 3t + t^2$.

It is significant to note that in order to find the minimal polynomial the MP algorithm only needed to compute the *rvec* of A and A^2 as opposed to the *rvec* of A, A^2, A^3 and A^4 which are necessary in the algorithm given in [1].

2.3 Some immediate applications of the MP algorithm

The following result is an immediate consequence of the MMP algorithm which is completely nontrivial to verify without the MMP algorithm.

Proposition 2.7 *A real matrix $A \in M_n(\mathbb{R})$ has a minimal polynomial with real coeficents.*

One application of the minimal polynomial is to obtain the inverse of a given matrix $A \in M_n(\mathbb{C})$ in the form of a matrix polynomial. Let $A \in M_n(\mathbb{C})$ be a given matrix. Then our algorithm computes the minimal polynomial, $q_A(t) = t^k + a_{k-1}t^{k-1} + \cdots + a_1 t + a_0$, $a_0 \neq 0$, of the matrix $A \in M_n(\mathbb{C})$. Thus, the inverse of the matrix $A \in M_n(\mathbb{C})$ is given by $A^{-1} = \frac{-1}{a_0}(A^{k-1} + a_{k-1}A^{k-2} + \cdots + a_1 I)$ as described in the introduction of this dissertation.

Example 2.4 We show how we may use the GU matrix to obtain the inverse of an invertible matrix. In Example (2.1), we obtained the minimal polynomial, $q_A(t) = -5 + 6t - 4t^2 + t^3$, for the invertible matrix

$$A = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 2 & 1 \\ 2 & 0 & 1 \end{bmatrix} \in M_3(\mathbb{C}).$$

When the MP algorithm terminated in example (2.1), we had the GU matrix

$$G_{\{v_0',v_1',v_2',v_3'\}}(B_4') = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & : & 2 & 0 & 0 & : & -2 & 2 & 1 & \| & 3 & -3 & 1 & 0 \\ 0 & 0 & 0 & : & 0 & 0 & 0 & : & 0 & 0 & 0 & \| & -5 & 6 & -4 & 1 \end{bmatrix}.$$

This means we can write the inverse using $A^{-1} = \dfrac{-1}{a_0}(a_3A^2 + a_2A + a_1I) = \dfrac{1}{5}(A^2 - 4A + 6I) = \dfrac{6}{5}I - \dfrac{4}{5}A + \dfrac{1}{5}A^2$. Instead of computing $\dfrac{1}{5}(A^2 - 4A + 6I) = \dfrac{6}{5}I - \dfrac{4}{5} + \dfrac{1}{5}A^2$, we use elementary row operations in order to have the coefficients $\frac{6}{5}$, $\frac{-4}{5}$, and $\frac{1}{5}$ appear in the third row of the GU matrix $G_{\{v_0',v_1',v_2',v_3'\}}(B_4')$. First, use the elementary operation $\dfrac{1}{5}R_3 \to R_3$ to obtain the GU matrix

$$G_{\{v_0',v_1',v_2',v_3'\}}(B_4') = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{5} & : & \frac{2}{5} & 0 & 0 & : & \frac{-2}{5} & \frac{2}{5} & \frac{1}{5} & \| & \frac{3}{5} & \frac{-3}{5} & \frac{1}{5} & 0 \\ 0 & 0 & 0 & : & 0 & 0 & 0 & : & 0 & 0 & 0 & \| & -5 & 6 & -4 & 1 \end{bmatrix}.$$

Next, we use the elementary operation $R_3 - \dfrac{1}{5}R_2 \to R_3$, which results in the GU matrix

$$G_{\{v_0',v_1',v_2',v_3'\}}(B_4') = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 & 0 & 0 \\ 0 & \frac{-1}{5} & \frac{1}{5} & : & \frac{3}{5} & \frac{-1}{5} & \frac{-1}{5} & : & \frac{-4}{5} & \frac{2}{5} & \frac{1}{5} & \| & \frac{4}{5} & \frac{-4}{5} & \frac{1}{5} & 0 \\ 0 & 0 & 0 & : & 0 & 0 & 0 & : & 0 & 0 & 0 & \| & -5 & 6 & -4 & 1 \end{bmatrix}.$$

Lastly, we use the elementary operation $R_3 + \dfrac{2}{5}R_1 \to R_3$, to obtain the GU matrix

$$G_{\{v_0',v_1',v_2',v_3'\}}(B_4') = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & : & -1 & 1 & 1 & : & 2 & 0 & 0 & \| & -1 & 1 & 0 & 0 \\ \frac{2}{5} & \frac{-1}{5} & \frac{1}{5} & : & \frac{3}{5} & \frac{1}{5} & \frac{-1}{5} & : & \frac{-4}{5} & \frac{2}{5} & \frac{3}{5} & \| & \frac{6}{5} & \frac{-4}{5} & \frac{1}{5} & 0 \\ 0 & 0 & 0 & : & 0 & 0 & 0 & : & 0 & 0 & 0 & \| & -5 & 6 & -4 & 1 \end{bmatrix}.$$

Notice that we can finally see the linear combination $\dfrac{6}{5}I - \dfrac{4}{5}A + \dfrac{1}{5}A^2$, and the inverse of matrix A is the matrix constructed by reversing the $rvec(\cdot)$ operation to get

$$A^{-1} = \frac{1}{5}\begin{bmatrix} 2 & -1 & 1 \\ 3 & 1 & -1 \\ -4 & 2 & 3 \end{bmatrix} \in M_3(\mathbb{C}).$$

In addition to finding the inverse of a matrix, the minimal polynomial also provides us with a way to write a matrix polynomial in its simplest possible form. The next example shows how we can use the MMP algorithm to obtain the matrix polynomial.

Example 2.5 Let

$$A = \begin{bmatrix} 3 & -1 & -1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{bmatrix} \in M_4(\mathbb{C})$$

and $p(t) = t^5 + 4t^3 + 11t + 27$ be a given polynomial. We will find the matrix for the matrix polynomial $p(A) = A^5 + 4A^3 + 11A + 27I$. We computed the minimal polynomial of A to be $q_A(t) = 2 - 3t + t^2$ using the MMP algorithm in Example (2.3). A calculation applying the Euclidean algorithm shows that $t^5 + 4t^3 + 11t + 27 = (t^3 + 3t^2 + 11t + 27)q_A(t) + (62t - 53)$. Thus, $p(A) = 62A - 53I$. In the last stage of Example (2.3), we obtained the GU matrix

$$G_{\{v_0', v_1', v_2'\}}(B_3') = \left[\begin{array}{cccccccccccccccccccccc} 1 & 0 & 0 & 0 & : & 0 & 1 & 0 & 0 & : & 0 & 0 & 1 & 0 & : & 0 & 0 & 0 & 1 & \| & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 & : & 1 & -2 & -1 & 0 & : & 1 & -1 & -2 & 0 & : & 1 & -1 & 0 & -2 & \| & -3 & 1 & 0 \\ 0 & 0 & 0 & 0 & : & 0 & 0 & 0 & 0 & : & 0 & 0 & 0 & 0 & : & 0 & 0 & 0 & 0 & \| & 2 & -3 & 1 \end{array}\right]$$

Using the elementary operation $62R_2 \to R_2$, the resulting GU matrix is

$$G_{\{v_0', v_1', v_2'\}}(B_3') = \left[\begin{array}{cccccccccccccccccccccc} 1 & 0 & 0 & 0 & : & 0 & 1 & 0 & 0 & : & 0 & 0 & 1 & 0 & : & 0 & 0 & 0 & 1 & \| & 1 & 0 & 0 \\ 0 & -62 & -62 & 0 & : & 62 & -124 & -62 & 0 & : & 62 & -62 & -124 & 0 & : & 62 & -62 & 0 & -124 & \| & -186 & 62 & 0 \\ 0 & 0 & 0 & 0 & : & 0 & 0 & 0 & 0 & : & 0 & 0 & 0 & 0 & : & 0 & 0 & 0 & 0 & \| & 2 & -3 & 1 \end{array}\right].$$

Next, use $R_2 + 133R_1 \to R_2$ to have the GU matrix

$$G_{\{v_0',v_1',v_2'\}}(B_3') = \left[\begin{array}{cccccccccccccccccccccc} 1 & 0 & 0 & 0 & : & 0 & 1 & 0 & 0 & : & 0 & 0 & 1 & 0 & : & 0 & 0 & 0 & 1 & \| & 1 & 0 & 0 \\ 133 & -62 & -62 & 0 & : & 62 & 9 & -62 & 0 & : & 62 & -62 & 9 & 0 & : & 62 & -62 & 0 & 9 & \| & -53 & 62 & 0 \\ 0 & 0 & 0 & 0 & : & 0 & 0 & 0 & 0 & : & 0 & 0 & 0 & 0 & : & 0 & 0 & 0 & 0 & \| & 2 & -3 & 1 \end{array}\right].$$

This shows us that the linear combination $-53I + 62A$ is

$$p(A) = \begin{bmatrix} 133 & -62 & -62 & 0 \\ 62 & 9 & -62 & 0 \\ 62 & -62 & 9 & 0 \\ 62 & -62 & 0 & 9 \end{bmatrix}$$

It is a well-known fact that an idempotent matrix, which is characterized by the property $A^2 = A$, has the minimal polynomial $q_A(t) = t(t-1)$. We use our MP algorithm to provide a new proof of this fact.

Proposition 2.8 *Let $A \in M_n(\mathbb{C})$ be an idempotent matrix different from the identity or zero matrix. Then A has minimal polynomial $q_A(t) = t(t-1)$.*

Proof. Let $A \in M_n(\mathbb{C})$ be an idempotent matrix other than the identity or zero matrix. Starting with $v_0' = [e_1^T \cdots e_n^T]$, we create the GU matrix $G_{\{v_0'\}}(I_1) = \left[\begin{array}{c|c} v_0' & 1 \end{array}\right]$. Following the algorithm, we compute $v_1 = v_0'(I \otimes A)$ and create the GU matrix

$$G_{\{v_0',v_1\}}(I_2) = \left[\begin{array}{c|cc} v_0' & 1 & 0 \\ v_1 & 0 & 1 \end{array}\right].$$

The resulting GU matrix from the row operation $R_2 - R_1 \to R_2$ is,

$$G_{\{v_0',v_1'\}}(B_2') = \left[\begin{array}{c|cc} v_0' & 1 & 0 \\ v_1' & -1 & 1 \end{array}\right].$$

We note that v_1' is the *rvec* of the matrix polynomial $-I + A$. Since A is not the identity $v_1' \neq 0$, and the algorithm continues. Computing $v_2 = v_1'(I \otimes A)$ and creating the GU matrix

$$G_{\{v_0',v_1',v_2\}}(I_2) = \left[\begin{array}{c|ccc} v_0' & 1 & 0 & 0 \\ v_1' & -1 & 1 & 0 \\ v_2 & 0 & -1 & 1 \end{array}\right],$$

we see that v_2 is the *rvec* of the matrix polynomial $-A + A^2$. Since A is idempotent we know that $-A + A^2 = 0$. Thus $v_2 = 0$, and the minimal polynomial of the idempotent matrix A is $q_A(t) = t(t-1)$.

□

A matrix is said to be nilpotent if $A^k = 0$ for some positive integer k. The smallest power k is called the index of nilpotency. The minimal polynomial of a nilpotent matrix is $q_A(t) = t^k$, where k is the index of nilpotency. Our algorithm will determine the nilpotency of a given matrix in the process of finding its minimal polynomial. Although finding the nilpotency of a matrix is a manageable task (one can compute the powers of the matrix to obtain the index of nilpotency), when we compute the minimal polynomial we get this information at a glance. It is usually difficult to recognize if a matrix is nilpotent by simple observation. Even for matrices of small size, nilpotency is not immediately detectible. For example, consider the 3×3 matrix

$$A = \begin{bmatrix} 5 & -3 & 2 \\ 15 & -9 & 6 \\ 10 & -6 & 4 \end{bmatrix} \in M_3(\mathbb{R}).$$

It is easy enough to compute and see that $A^2 = 0$. It is nice to know that in the process of calculating the minimal polynomial we obtain the same information.

Example 2.6

In this example we calculate the minimal polynomial for the matrix

$$A = \begin{bmatrix} 5 & -3 & 2 \\ 15 & -9 & 6 \\ 10 & -6 & 4 \end{bmatrix} \in M_3(\mathbb{R}).$$

The matrix will be shown to be nilpotent in the process of computing its minimal polynomial.

Starting with the vector $v_0' = rvec(I_3) = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 \end{bmatrix}$, we construct the GU matrix

$$G_{\{v_0'\}}(B_1) = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 \end{bmatrix}.$$

Next we compute

$$\begin{aligned} v_1 &= v_0'(I \otimes A) \\ &= \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{bmatrix} \\ &= \begin{bmatrix} 5 & -3 & 2 & : & 15 & -9 & 6 & : & 10 & -6 & 4 \end{bmatrix}. \end{aligned}$$

Now wev construct the GU matrix

$$G_{\{v_0',v_1\}}(B_2) = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 \\ 5 & -3 & 2 & : & 15 & -9 & 6 & : & 10 & -6 & 4 & \| & 0 & 1 \end{bmatrix}.$$

Using the elementary row operation $R_2 - 5R_1 \to R_2$, we obtain the GU matrix

$$G_{\{v_0',v_1'\}}(B_2') = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 \\ 0 & -3 & 2 & : & 15 & -14 & 6 & : & 10 & -6 & -1 & \| & -5 & 1 \end{bmatrix}.$$

Since $v_1' \neq 0$, we compute

$$\begin{aligned} v_2 &= v_1'(I \otimes A) \\ &= \begin{bmatrix} 0 & -3 & 2 & : & 15 & -14 & 6 & : & 10 & -6 & -1 \end{bmatrix} \begin{bmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{bmatrix} \\ &= \begin{bmatrix} -25 & 15 & 10 & : & -75 & 45 & -30 & : & -50 & 30 & -20 \end{bmatrix}. \end{aligned}$$

and construct the GU matrix $G_{\{v_0', v_1', v_2\}}(B_3) =$

$$\begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 \\ 0 & -3 & 2 & : & 15 & -14 & 6 & : & 10 & -6 & -1 & \| & -5 & 1 & 0 \\ -25 & 15 & 10 & : & -75 & 45 & -30 & : & -50 & 30 & -20 & \| & 0 & -5 & 1 \end{bmatrix}.$$

Using the elementary row opertations $R_3 + 25R_1 \to R_3$ and $R_3 + 5R_2 \to R_3$, we end up with the GU matrix $G_{\{v_0', v_1', v_2'\}}(B_3') =$

$$\begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 0 & : & 0 & 0 & 1 & \| & 1 & 0 & 0 \\ 0 & -3 & 2 & : & 15 & -14 & 6 & : & 10 & -6 & -1 & \| & -5 & 1 & 0 \\ 0 & 0 & 0 & : & 0 & 0 & 0 & : & 0 & 0 & 0 & \| & 0 & 0 & 1 \end{bmatrix}.$$

Therefore, the minimal polynomial of the matrix A is $q_A(t) = t^2$, which shows that the matrix A is nilpotent with nilpotency two.

CHAPTER 3

Applications and observations of the modified minimal polynomial algorithm (MMP)

Now that we have shown that the MMP algorithm will find the minimal polynomial of any given matrix $A \in M_n(\mathbb{C})$, this chapter studies how the MMP algorithm can be effectively implemented to matrices of special structure. First, we show that the MMP algorithm can be significantly simplified when it is applied to an unreduced lower Hessenberg matrix. Then, we use the MMP algorithm to obtain a recursive formula for the minimal polynomial for a tridiagonal matrix. We conclude this chapter with further applications of our algorithm.

3.1 Lower Hessenberg matrices

There are classes of matrices that we examine in this chapter in order to see the effect a matrices structure has on the computation of its minimal polynomial. One such class is that of lower Hessenberg matrices, and so we include its definition.

Definition 3.1 *A matrix $A = [a_{ij}] \in M_n(\mathbb{C})$ is said to be a lower-Hessenberg matrix if $a_{ij} = 0$ for $j > i + 1$.*

Lower Hessenberg matrices have the form

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \ddots & \ddots & \vdots \\ a_{31} & a_{32} & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & a_{n-1,n} \\ a_{n1} & a_{n2} & \cdots & a_{n,n-1} & a_{nn} \end{bmatrix}.$$

A lower-Hessenberg matrix is said to be unreduced if all the superdiagonal entries are non-zero. In other words, $a_{i,i+1} \neq 0$ for each $i = 1, \ldots, n-1$. If there are zero entries on the superdiagonal, the lower-Hessenberg matrix is said to be reduced.

Proposition 3.1 *Let $A \in M_n(\mathbb{C})$ be an unreduced lower Hessenberg matrix. Then the first row of each matrix in the set $\{I, A, A^2, \ldots, A^{n-1}\}$ forms a basis for $\mathbb{C}^n$.*

Proof. Let

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \ddots & \ddots & \vdots \\ a_{31} & a_{32} & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & a_{n-1,n} \\ a_{n1} & a_{n2} & \cdots & a_{n,n-1} & a_{nn} \end{bmatrix} \in M_n(\mathbb{C})$$

be an unreduced lower-Hessenberg matrix. Then $a_{i,i+1} \neq 0$ for $i = 1, \ldots, n-1$. We observe that

$$A^2 = \begin{bmatrix} a_{11}^{(2)} & a_{12}^{(2)} & a_{13}^{(2)} & \cdots & 0 \\ a_{21}^{(2)} & a_{22}^{(2)} & \ddots & \ddots & \vdots \\ a_{31}^{(2)} & a_{32}^{(2)} & \ddots & \ddots & a_{n-2,n}^{(2)} \\ \vdots & \vdots & \ddots & \ddots & a_{n-1,n}^{(2)} \\ a_{n1}^{(2)} & a_{n2}^{(2)} & \cdots & a_{n,n-1}^{(2)} & a_{nn}^{(2)} \end{bmatrix}$$

where $a_{i,i+2}^{(2)} \neq 0$ for $i = 1, \ldots, n-2$, and in general we see that

$$A^k = \begin{bmatrix} a_{11}^{(k)} & a_{12}^{(k)} & a_{13}^{(k)} & \cdots & a_{1,1+k}^{(k)} & 0 & 0 \\ a_{21}^{(k)} & a_{22}^{(k)} & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_{31}^{(k)} & a_{32}^{(k)} & \ddots & \ddots & \ddots & \ddots & a_{n-k,n}^{(k)} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & a_{n-2,n}^{(k)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & a_{n-1,n}^{(k)} \\ a_{n1}^{(k)} & a_{n2}^{(k)} & \cdots & \cdots & \cdots & a_{n,n-1}^{(k)} & a_{nn}^{(k)} \end{bmatrix},$$

where $a_{i,i+k}^{(k)} \neq 0$ for $i = 1, \ldots, n-k$.

A consequence of this observation is that the set

$$\Big\{(1, 0, \ldots, 0), (a_{11}, a_{12}, 0, \ldots, 0), \ldots, (a_{11}^{(n)}, a_{12}^{(n)}, \ldots, a_{1n}^{(n)})\Big\}$$

is linearly independent. This means that the first rows in the matrices $I, A, A^2, \ldots,$ and A^{n-1} form a linearly independent set. This set also spans $\mathbb{C}^n$, which shows it is a basis. $\square$

A simple consequence of Proposition (3.1) is the following theorem.

Theorem 3.1 *Let $A \in M_n(\mathbb{C})$ be an unreduced lower-Hessenberg matrix. Then the MP algorithm only requires the first rows of the matrices $I, A, A^2, \ldots, A^{n-1}$ to compute the minimal polynomial of $A \in M_n(\mathbb{C})$. In this case, the minimal polynomial is equal to the characteristic polynomial of the matrix A.*

Proof. Let $A \in M_n(\mathbb{C})$ be an unreduced lower Hessenberg matrix. Then the first rows of $\{I, A, A^2, \ldots, A^{n-1}\}$ forms a basis for $\mathbb{C}^n$ by Proposition (3.1). In this case, the MP algorithm requires $n+1$ steps to complete by using only the first row of the matrices to obtain the minimal polynomial of the A. Since the minimal polynomial is degree n, the minimal polynomial must be the characteristic polynomial of A. $\square$

Theorem (3.1) shows that the MMP algorithm is significantly simplified when it is applied to lower Hessenberg matrices since the algorithm needs to compute only the first row of the matrices $A, A^2, \ldots$ and A^{n-1}. This means that instead of computing $v_k = v'_{k-1}(I \otimes A)$, where $v_i \in \mathbb{C}_{n^2}$, we only need to compute $v_k = v'_{k-1}A$ where $v_i \in \mathbb{C}_n$. This simplification reduces the amount of computation from the order of n^2 to the order of n. The algorithm below shows how the MMP algorithm is applied to lower Hessenberg matrices.

The MMP Algorithm for Lower Hessenberg Matrices

For an unreduced lower Hessenberg matrix $A \in M_n(\mathbb{C})$, let v_i be the first row of the matrix $A^{(i)}$, $A^{(0)} = I_n$, and do the following.

Step 1. Create $G_{\{v_0\}}(I_1)$, set $v_0 = e_1^T \equiv v'_0$, where $e_1^T \in \mathbb{C}_n$, set $i = 1$, and $B_1 \equiv I_1$.

Step 2. Compute $v_i = v'_{i-1}A$ and construct the GU matrix $G_{\{v'_0,\ldots,v'_{i-1},v_i\}}(B_{i+1})$ where $B_{i+1} \equiv \begin{bmatrix} B'_i & 0 \\ 0 & b \end{bmatrix}$, such that $b \in \mathbb{C}_i$ are the entries of the last row of B'_i. Use Gaussian elimination to obtain $G_{\{v'_0,\ldots,v'_{i-1},v'_i\}}(B'_{i+1})$.

- If $v'_i \equiv 0$ stop and proceed to Step 3.
- If $v'_i \neq 0$, increment i by 1 and repeat Step 2.

Step 3. For $i = k$ such that $v'_k \equiv 0$, the entries of the last row of B'_{k+1}, $b_{k+1,j} \in \mathbb{C}$ with $j = 1, \ldots, k+1$, are the coefficients of the minimal polynomial of the matrix $A \in M_n(\mathbb{C})$.

We now give an example to illustrate how Theorem (3.1) can be applied.

Example 3.1 Consider

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 0 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \in M_4(\mathbb{R}).$$

We start the MMP algorthim for lower Hessenberg matrices with $v_0' = [1\,0\,0\,0]$, and we construct the GU matrix

$$G_{\{v_0'\}}(I_1) = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 1 \end{array}\right].$$

Next, we compute

$$v_1 = v_0' A = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}$$

and construct the GU matrix

$$G_{\{v_0', v_1\}}(I_2) = \left[\begin{array}{cccc|cc} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right].$$

Now we check whether v_0' and v_1 are linearly dependent using the elementary row operation $R_2 - R_1 \to R_1$. The result is the GU matrix

$$G_{\{v_0', v_1'\}}(B_2') = \left[\begin{array}{cccc|cc} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 & 1 \end{array}\right].$$

Since $v_1' \neq 0$, the vectors v_0' and v_1 are not linearly dependent. The algorithm continues by computing

$$v_2 = v_1' A = \begin{bmatrix} 2 & 1 & 1 & 0 \end{bmatrix},$$

and constructing the GU matrix

$$G_{\{v_0', v_1', v_2\}}(B_3) = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 1 & 0 \\ 2 & 1 & 1 & 0 & 0 & -1 & 1 \end{array}\right].$$

Using the elementary row operations $R_3 - 2R_1 \to R_3$ and $R_3 - R_1 \to R_3$, the resulting GU matrix is

$$G_{\{v_0',v_1',v_2'\}}(B_3') = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & -2 & 1 \end{array}\right].$$

We see that $v_2' \neq 0$, and so the vectors v_0', v_1' and v_2 are not linearly dependent. The algorithm continues by computing

$$v_3 = v_2'A = \left[\begin{array}{cccc} 1 & 2 & 3 & 1 \end{array}\right]$$

and constructing the GU matrix

$$G_{\{v_0',v_1',v_2',v_3\}}(B_4) = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & -2 & 1 & 0 \\ 1 & 2 & 3 & 1 & 0 & -1 & -2 & 1 \end{array}\right].$$

Applying Gaussian elimination using the elementary row operations $R_4 - R_1 \to R_4, R_4 - 2R_2 \to R_4$, and $R_4 - 3R_3 \to R_4$, we obtain the GU matrix

$$G_{\{v_0',v_1',v_2',v_3'\}}(B_4') = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 4 & 3 & -5 & 1 \end{array}\right].$$

The vectors v_0', v_1', v_2' and v_3 are again not linearly dependent, and the algorithm continues. We compute

$$v_4 = v_3'A = \left[\begin{array}{cccc} 1 & 1 & 1 & 1 \end{array}\right]$$

and construct the GU matrix

$$G_{\{v_0',v_1',v_2',v_3',v_4\}}(B_5) = \left[\begin{array}{cccc|ccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & 3 & -5 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 4 & 3 & -5 & 1 \end{array}\right].$$

The row operations $R_5 - R_1 \to R_5, R_5 - R_2 \to R_5, R_5 - R_3 \to R_5$ and $R_5 - R_4 \to R_5$ are used and the resulting GU matrix is

$$G_{\{v_0', v_1', v_2', v_3', v_4'\}}(B_5') = \left[\begin{array}{cccc|ccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & 3 & -5 & 1 & 0 \\ 0 & 0 & 0 & 0 & -3 & 2 & 7 & -6 & 1 \end{array}\right].$$

Since $v_4' = 0$ the algorithm terminates, and the coefficients of the minimal polynomial can be read off from the last row of the matrix B_5'. The minimal polynomial of the matrix A is $q_A(t) = -3 + 2t + 7t^2 - 6t^3 + t^4$.

A simple consequence of Theorem (3.1) is that any monic polynomial, $p(t)$, is both the minimal polynomial and the characteristic polynomial of its companion matrix. The companion matrix of a monic polynomial $p(t)$ is defined as follows.

Definition 3.2 *Let $p(t) = t^n + a_{n-1}t^{n-1} + \cdots + a_1 t + a_0$. A matrix $A = [a_{ij}] \in M_n(\mathbb{C})$ is a* ***companion matrix*** *of the polynomial $p(t)$ if it has the form*

$$A = \begin{bmatrix} 0 & 1 & & 0 \\ \vdots & 0 & 1 & \\ 0 & & \ddots & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix}$$

The next theorem is an immediate consequence of Theorem (3.1).

Theorem 3.2 *Every monic polynomial is both the minimal polynomial and the characteristic polynomial of its companion matrix.*

Theorem (3.1) required that the lower-Hessenberg matrix be in unreduced form; it depended on the superdiagonal elements being non-zero. Sometimes, however, there are zeros on the superdiagonal. In that case, the matrix is in unreduced form. In this situation, the MMP algorithm for lower Hessenberg matrices will compute the characteristic polynomial which may not coincide with the minimal polynomial. The minimal polynomial can always be obtained by implementing the MMP algorithm without reduction to a lower Hessenberg matrices. We have the following result.

Theorem 3.3 *Let $A \in M_n(\mathbb{C})$ be a block lower triangular form,*

$$A = \begin{bmatrix} H_1 & & & \mathit{0} \\ & H_2 & & \\ & & \ddots & \\ * & & & H_k \end{bmatrix}$$

where $H_i \in M_{n_i}(\mathbb{C})$, $i = 1, \ldots, k$, $n_1 + n_2 + \cdots + n_k = n$, are unreduced lower Hessenberg matrices. For each i, let $p_i(t)$ be the minimal polynomial of the unreduced lower Hessenberg matrix $H_i \in M_{n_i}$, respectively. Then, $p_A(t) = p_1(t) \cdots p_k(t)$ is the characteristic polynomial of A.

Proof. Let A be a matrix of the form in the statement of the theorem. We observe that the characteristic polynomial of A is $p_A(t) = det(A - tI)) = det(H_1 - tI) \cdots det(H_k - tI)$ [14, p. 205]. Since each H_i is in unreduced lower Hessenberg form, the MMP algorithm computes the minimal polynomial of each H_i which is the characteristic polynomial of each H_i by Theorem (3.1). Thus, the product $det(H_1 - tI) \cdots det(H_k - tI)$, for $i = 1, \ldots, k$ is an n-th degree polynomial that annihilates A, which is the characteristic polynomial of A. □

Example 3.2 Let

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 5 & 3 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 1 & 3 & 3 & 0 & 0 \\ 2 & 3 & 5 & 4 & 7 & 9 & 1 & 1 \\ 2 & 3 & 1 & 5 & 8 & 10 & 1 & 1 \end{bmatrix} = \begin{bmatrix} H_1 & & 0 \\ & H_2 & \\ * & & H_3 \end{bmatrix} \in M_8(\mathbb{R}),$$

where diagonal blocks $H_1, H_2 \in M_3(\mathbb{R})$, and $H_3 \in M_2(\mathbb{R})$ are lower Hessenberg matrices. Implementing the algorithm we obtain $q_{H_1}(t) = t(1 - 3t + t^2)$, $q_{H_2}(t) = 5 + 2t - 5t^2 + t^3$, and $q_{H_3}(t) = t(-2 + t)$. Thus the characteristic polynomial of A is $p_A(t) = t^2(t-2)(t^3 - 5t^2 + 2t + 5)(t^2 - 3t + 1)$.

Any matrix $A \in M_n(\mathbb{C})$ can be transformed into a lower Hessenberg matrix under the similarity transformation. Theorems (3.1) and (3.3) now give us a way to obtain the characteristic polynomial of any matrix $A \in M_n(\mathbb{C})$ using the MMP algorithm on lower Hessenberg matrices. Applying the MMP algorithm to the resulting lower Hessenberg matrix, either in unreduced or reduced form, results in the minimal or the characteristic polynomial, respectively.

3.2 Hermitian Matrices

In this section, we show how the MMP algorithm can be easily applied to obtain the minimal polynomial of a Hermitian matrix.

Definition 3.3 *A matrix $A = [a_{ij}] \in M_n(\mathbb{C})$ is said to be Hermitian if $A = A^*$, where $A^* \equiv \overline{A}^T = [\overline{a}_{ji}]$.*

Hermitian matrices are unitarily real diagonalizable. Since this is the case we know that the minimal polynomial of a Hermitian matrix $A \in M_n(\mathbb{C})$ has the form $q_A(t) = (t - \lambda_1) \cdots (t - \lambda_k)$, where the λ_i's, $i = 1, \ldots, k$ are the distinct eigenvalues of the matrix A. Since the roots of the minimal polynomial of any matrix contains all the distinct eigenvalues of the matrix we have the following application of the MP algorithm.

Application 3.1 The MP algorithm can be used to count the number of distinct eigenvalues of a Hermitian matrix. More generally, the MP algorithm will count the distinct eigenvalues of any diagonalizable matrix.

If $A \in M_n(\mathbb{C})$ is diagonalizable, the $rank(A)$ counts the number of non-zero eigenvalues of the matrix. However, there is no easy way of counting the number of distinct eigenvalues of a diagonalizable matrix. The significance of the Application (3.1) is that it tells us the number of distinct eigenvalues of a real diagonalizable matrix once the minimal polynomial is computed.

A real Hermitian matrix is known to be orthogonally similar to a tridiagonal matrix via Givens planar rotations. We will describe a process of how to find a tridiagonal matrix that is similar to a given Hermitian matrix. Then we will exploit the structure of a tridiagonal matrix in order to find a recursive formula for the minimal polynomial of a real Hermitian matrix.

Definition 3.4 ***(Givens Matrix)*** *Let*

$$U(\theta,i,j)=\left[\begin{array}{ccccccccccc}
1 & & & 0 & & & & 0 & & & \\
 & \ddots & & \vdots & & 0 & & \vdots & & 0 & \\
 & & 1 & 0 & & & & 0 & & & \\
0 & \cdots & 0 & \cos(\theta) & 0 & \cdots & 0 & \sin(\theta) & 0 & \cdots & \\
 & & & 0 & 1 & & & 0 & & & \\
 & 0 & & \vdots & & \ddots & & \vdots & & 0 & \\
 & & & 0 & 0 & & 1 & 0 & & & \\
0 & \cdots & 0 & -\sin(\theta) & 0 & \cdots & 0 & \cos(\theta) & 0 & \cdots & \\
 & & & 0 & & & & 0 & 1 & & \\
 & 0 & & \vdots & & 0 & & \vdots & & \ddots & \\
 & & & 0 & & & & 0 & & & 1
\end{array}\right].$$

This is the identity matrix with the (i,i) and (j,j) entries replaced by $\cos(\theta)$ and the (i,j) and (j,i) entries replaced by $\sin(\theta)$ and $-\sin(\theta)$, respectively.

It is clear that any Givens matrix is unitary. The Givens matrices can be used to place zeros in any entry in a vector or a matrix. The following lemma is our first step in using the Givens matrices to tridiagonalize a real Hermitian matrix.

Lemma 3.1 *Suppose $A=\begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22}\end{bmatrix}\in M_2(\mathbb{R})$ is a given real Hermitian matrix. Then, $B=\begin{bmatrix} c & s \\ -s & c\end{bmatrix}\in M_2(\mathbb{R})$, where $c=\sqrt{\frac{1}{2}-\sqrt{\frac{1}{4}-\omega}}$, $s=\sqrt{\frac{1}{2}+\sqrt{\frac{1}{4}-\omega}}$, and $\omega=\dfrac{1}{4+(\frac{a_{22}-a_{11}}{a_{12}})^2}$ a Givens matrix such that $BAB^*=\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2\end{bmatrix}$, $\lambda_i\in\mathbb{R}, i=$* $1,2$.

Proof. Suppose $A=\begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22}\end{bmatrix}\in M_2(\mathbb{R})$ is a given real Hermitian matrix. If $a_{12}=0$ then the matrix is already in the desired form. For this reason we may assume that $a_{12}\neq 0$. The matrix $B=\begin{bmatrix} c & s \\ -s & c\end{bmatrix}$ is a Givens matrix because $c^2+s^2=\left(\sqrt{\frac{1}{2}-\sqrt{\frac{1}{4}-\omega}}\right)^2+\left(\sqrt{\frac{1}{2}+\sqrt{\frac{1}{4}-\omega}}\right)^2=1$, when $\omega=\dfrac{1}{4+(\frac{a_{22}-a_{11}}{a_{12}})^2}$.

Now, upon matrix multiplication, we have

$$BAB^* = \begin{bmatrix} \lambda_1 & (c^2-s^2)a_{22} + cs(a_{12}-a_{11}) \\ (c^2-s^2)a_{22} + cs(a_{12}-a_{11}) & \lambda_2 \end{bmatrix},$$

where $\lambda_1 = c^2a_{11}+s^2a_{22}+2sca_{12},\ \lambda_2 = c^2a_{22}+s^2a_{11}-2sca_{12} \in \mathbb{R}$. Since $s^2+c^2=1$, $c = \sqrt{\frac{1}{2} - \sqrt{\frac{1}{4}-\omega}}$, $s = \sqrt{\frac{1}{2} + \sqrt{\frac{1}{4}-\omega}}$, and $\omega = \dfrac{1}{4+(\frac{a_{22}-a_{11}}{a_{12}})^2}$.

□

Note that if $a_{11} = a_{22}$ in the Lemma (3.1) then $s = c = \dfrac{1}{\sqrt{2}}$ and the Givens matrix becomes $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$. Since Givens matrices only effect the i-th and j-th rows and columns during multiplication, Lemma (3.1) allows us to tridiagonalize a real Hermitian matrix by successively introducing zeros in the off-diagaonal entries of the Hermitian matrix. For example, consider the 4×4 real Hermitian matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix}.$$

Using the Givens matrix $U(\theta,1,4)$ and choosing θ in accordance with the previous lemma , we may place a zero in the a_{14} element of matrix A by computing $U(\theta,1,4)AU^*(\theta,1,4)$. This results in the real Hermitian matrix

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & 0 \\ b_{12} & b_{22} & b_{23} & b_{24} \\ b_{13} & b_{23} & b_{33} & b_{34} \\ 0 & b_{24} & b_{34} & b_{44} \end{bmatrix} = U(\theta,1,4)AU^*(\theta,1,4).$$

Now we would like to place a zero in the b_{24} entry of this real Hermitian matrix through the use of the Givens matrix $U(\theta, 2, 3)$ along with θ determined in Lemma (3.1) by computing $U(\theta, 2, 3)BU^*(\theta, 2, 3)$. This multiplication gives us real Hermitian matrix

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & 0 \\ c_{12} & c_{22} & c_{23} & 0 \\ c_{13} & c_{23} & c_{33} & c_{34} \\ 0 & 0 & c_{34} & c_{44} \end{bmatrix} = U(\theta, 2, 3)BU^*(\theta, 2, 3).$$

Notice that the zero in the $(1, 4)$ entry is not disturbed by the multiplication of this Givens matrix . As a final step in the tridiagonal process, we will use the Givens matrix $U(\theta, 1, 2)$ and θ deteremined from Lemma (3.1) to place a zero in the c_{13} entry of the matrix C. We multiply $U(\theta, 1, 2)CU^*(\theta, 1, 2)$ to obtain a real Hermitian matrix

$$D = \begin{bmatrix} d_{11} & d_{12} & 0 & 0 \\ d_{12} & d_{22} & d_{23} & 0 \\ 0 & d_{23} & d_{33} & d_{34} \\ 0 & 0 & d_{34} & d_{44} \end{bmatrix} = U(\theta, 1, 2)CU^*(\theta, 1, 2).$$

which is tridiagonal. We note again that the $(1, 4)$ and $(2, 4)$ entries of the matrix, which the process had previously turned into zeros, were not disturbed by the matrix multiplication. Thus, the Hermitian matrix A has been tridiagonalized by successively applying sequences of Givens matrices.

This process can easily be generalized to tridiagonalize any real Hermitian matrix. The following example is included to illustrate the process of triadiagonalizing a real Hermitian matrix using Givens matrices.

Example 3.3 Consider

$$A = \begin{bmatrix} 15 & \frac{7}{\sqrt{2}} & 13\sqrt{2} & -5 \\ \frac{7}{\sqrt{2}} & \frac{-58}{5} & \frac{31}{5} & \frac{17}{\sqrt{2}} \\ 13\sqrt{2} & \frac{31}{5} & \frac{-92}{5} & 3\sqrt{2} \\ -5 & \frac{17}{\sqrt{2}} & 3\sqrt{2} & 15 \end{bmatrix} \in M_4(\mathbb{C}).$$

Then A is a real Hermitian matrix. To place a zero in the $(1,4)$ entry we will use the Givens matrix $U(\theta,1,4)$ where $c = s = \frac{1}{\sqrt{2}}$ by Lemma (3.1). Thus, the Givens matrix is

$$U(\theta,1,4) = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

Multilying with this Givens matrix, we obtain the real Hermitian matrix

$$U^*(\theta,1,4)AU(\theta,1,4) = \begin{bmatrix} 20 & -5 & 10 & 0 \\ -5 & \frac{-58}{5} & \frac{31}{5} & 12 \\ 10 & \frac{31}{5} & \frac{-92}{5} & 16 \\ 0 & 12 & 16 & 10 \end{bmatrix} = B.$$

We now choose $U(\theta,2,3)$ to obtain a zero in the $(2,4)$ entry in the matrix B. Using Lemma (3.1), we easliy compute $c = \frac{4}{5}$ and $s = \frac{3}{5}$ to obtain the Givens matrix

$$U(\theta,2,3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{4}{5} & \frac{3}{5} & 0 \\ 0 & -\frac{3}{5} & \frac{4}{5} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$U(\theta, 2, 3)$ and $U(\theta, 2, 3)^*$ are multiplied to B to result in the real Hermitian matrix

$$U^*(\theta, 2, 3)BU(\theta, 2, 3) = \begin{bmatrix} 20 & -10 & 5 & 0 \\ -10 & -20 & 5 & 0 \\ 5 & 5 & -10 & 20 \\ 0 & 0 & 20 & 10 \end{bmatrix} = C.$$

We finish the tridiagonalization of matrix A using $U(\theta, 1, 2)$ to force a zero in the $(1, 3)$ entry of matrix C. Using Lemma (3.1) we compute $c = s = \dfrac{1}{\sqrt{2}}$ to obtain the Givens matrix

$$U(\theta, 1, 2) = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Multiplying with this choice of Givens matrix results in the real Hermitian matrix

$$U^*(\theta, 1, 2)CU(\theta, 1, 2) = \begin{bmatrix} 10 & 20 & 0 & 0 \\ 20 & -10 & 5\sqrt{2} & 0 \\ 0 & 5\sqrt{2} & -10 & 20 \\ 0 & 0 & 20 & 10 \end{bmatrix},$$

which is unitarily similar to matrix A. The motivation for tridiagonalizing a Hermitian matrix is to apply the MMP algorithm for lower Hessenberg matrices to a tridiagonal matrix. Since a tridiagonal matrix is a special type of lower Hessenberg matrix, we recall that the MMP algorithm is significantly simplified by only using the first row of the matrices $I, A, \ldots,$ and A^{n-1}. Moreover, in the case of a tridiagonal matrix, the MMP algorithm for lower Hessenberg matrices actually produces a simple recurrence formula for the minimal polynomial.

We show how a recurrence relation is easily obtained from the MMP algorithm in the case of a tridiagonal matrix. Let $p_i(t)$ be the polynomial obtained from i-th row of the augmented matrix B_{k+1} in the GU matrix $G_{\{v_0', v_1, \ldots, v_k\}}(B_{k+1})$. First we observe that $p_0(t) = 1$ and $p_1(t) = t - a_1$. Then, we notice that the last row of the augmented matrix B_{k+2} in the GU matrix $G_{\{v_0', v_1', \ldots, v_k', v_{k+1},\}}(B_{k+2})$ is obtained by shifting the last row of B_{k+1} in $G_{\{v_0', v_1', \ldots, v_{k-1}', v_k,\}}(B_{k+1})$ by one column to the right and that corresponds to multiplying the polynomial $p_{k-1}(t)$ by t in the GU matrix $G_{\{v_0', v_1', \ldots, v_k'\}}(B_{k+1})$. Then the simple structure of the tridiagonal matrix allows us to obtain the last row of the augmented matrix B_{k+2}' by using the k-th and $(k+1)$-th rows of B_{k+2}. This observation gives the following remark.

Remark 3.1 *Let*

$$A = \begin{bmatrix} a_1 & b_1 & & \mathit{0} \\ c_1 & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ \mathit{0} & & c_{n-1} & a_n \end{bmatrix} \in M_n(\mathbb{C}),$$

where $b_i \neq 0$ for $i = 1, \ldots, n-1$. Then the minimal polynomial of the tridiagonal matrix is obtained from the following recursive formula $p_0(t) = 1, p_1(t) = t - a_1$ and $p_k(t) = (t - a_k)p_{k-1}(t) - b_{k-1}c_{k-1}p_{k-2}(t)$ for $k = 2, \ldots, n$.

The recursive formula for a tridiagonal matrix can be reduced further for special types of tridiagonal matrices. A very specific example of this is the tridiagonal Toeplitz matrix which has the form

$$A = \begin{bmatrix} a & b & & 0 \\ c & \ddots & \ddots & \\ & \ddots & \ddots & b \\ 0 & & c & a \end{bmatrix} \in M_n(\mathbb{C}).$$

In this case, the recursive formula may be reduced to $p_0(t) = 1, p_1(t) = t - a$ and $p_k(t) = (t-a)p_{k-1}(t) - bcp_{k-2}(t)$ for $k = 2, \ldots, n$.

Example 3.4 In Example (3.3) we reduced the matrix

$$A = \begin{bmatrix} 15 & \frac{7}{\sqrt{2}} & 13\sqrt{2} & -5 \\ \frac{7}{\sqrt{2}} & \frac{-58}{5} & \frac{31}{5} & \frac{17}{\sqrt{2}} \\ 13\sqrt{2} & \frac{31}{5} & \frac{-92}{5} & 3\sqrt{2} \\ -5 & \frac{17}{\sqrt{2}} & 3\sqrt{2} & 15 \end{bmatrix} \in M_4(\mathbb{C})$$

to the unitarily similar tridiagonal matrix

$$D = \begin{bmatrix} 10 & 20 & 0 & 0 \\ 20 & -10 & 5\sqrt{2} & 0 \\ 0 & 5\sqrt{2} & -10 & 20 \\ 0 & 0 & 20 & 10 \end{bmatrix} \in M_4(\mathbb{C}).$$

Instead of computing the minimal polynomial of matrix A using the MMP algorithm, we will take advantage of the fact that the minimal polynomial of similar matrices are equal. We compute the minimal polynomial of the tridiagonal matrix D using the recursive formula in Remark (3.1). We start with $p_0(t) = 1$ and $p_1(t) = t - 10$. Then use Remark (3.1)

$$\begin{aligned} p_2(t) &= (t-a_2)p_1(t) - b_1c_1p_0(t) \\ &= t^2 - 500, \\ p_3(t) &= (t-a_3)p_2(t) - b_2c_2p_1(t) \\ &= t^3 + 10t^2 - 550t - 4500, \\ p_4(t) &= (t-a_4)p_3(t) - b_3c_3p_2(t) \\ &= t^4 - 1050t^2 + 1000t + 245000 \end{aligned}$$

which is the minimal polynomial of matrix D, and hence of A. The minimal polynomial of matrix A is $q_A(t) = t^4 - 1050t^2 + 1000t + 245000$.

The following theorem is a well-known result about the eigenvalues of tridiagonal matrices [3, Lemma 0.1.1, p. 7]. We give a new proof of the theorem using the MP algorithm in the next chapter.

Theorem 3.4 *Let*

$$A = \begin{bmatrix} a_1 & b_1 & & \mathit{0} \\ c_1 & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ \mathit{0} & & c_{n-1} & a_n \end{bmatrix} \in M_n(\mathbb{R})$$

be a tridiagonal matrix with $b_i c_i > 0$ *for* $i = 1, \ldots, n-1$. *Then the eigenvalues of the matrix* A *are real and distinct.*

CHAPTER 4

Further applications of the MP algorithm

This chapter gives further applications of the MMP algorithm. First, we give a systematic method to determine if a matrix is or not real diagonalizable. Diagonalizable matrices arise in many areas of matrix theory, and the assumption of diagonalizability is a minimum requirement in most studies where square matrices are employed. Therefore, it is important to know when a given matrix is (real) diagonalizable. Also in this chapter, we define a new class of real diagonalizable matrices, called the pre-Hermitian matrices. We provide basic properties of the pre-Hermitian matrices and show how we may apply the MMP algorithm to compute the minimal polynomial of a pre-Hermitian matrix.

4.1 Diagonalization

A matrix $A \in M_n(\mathbb{C})$ is said to be diagonalizable if there is a nonsingular matrix $R \in M_n(\mathbb{C})$ such that $R^{-1}AR = \Lambda$, where $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$, $\lambda_i \in \sigma(A)$. A diagonalizable matrix is said to be real diagonalizable if all of its eigenvalues are real. A matrix $A \in M_n(\mathbb{C})$ is unitarily diagonalizable if there is a unitary matrix $U \in U_n(\mathbb{C})$ such that $U^{-1}AU = \Lambda$, where $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$, $\lambda_i \in \sigma(A)$. The following result about diagonalizable matrices is well known.

Theorem 4.1 *[7, Theorem 1.3.7] A matrix $A \in M_n(\mathbb{C})$ is diagonalizable if and only if there exists a set of n linearly independent eigenvectors of A.*

Theorem (4.1) seems to completely characterize the diagonaliability of a complex matrix. Theoretically at least, all one has to do is to determine whether a given matrix has n linearly independent eigenvectors. However, computing all possible linearly independent eigenvectors for a given matrix is a complicated matter.

Another well known result views the diagonalization of a matrix in an entirely different way. The result relies on the Cayley-Hamilton theorem in which the diagonalizability is determined by simple additions and multiplications of matrices. Hence, the result is computationally simpler to apply.

Theorem 4.2 *[7, Corollary 3.3.8] A matrix $A \in M_n(\mathbb{C})$ is diagonazable if and only if the minimal polynomial of A splits into distinct linear factors. In other words, $q_A(t) = (t - \lambda_1) \cdots (t - \lambda_k)$ where $\lambda_1, \ldots, \lambda_k$ are the distinct eigenvalues of A.*

This result along with the MMP algorithm can be applied to determine whether a matrix $A \in M_n(\mathbb{C})$ is diagonalizable. Once the mimimal polynomial of $A \in M_n(\mathbb{C})$ is obtained by use of the MMP algorithm, we determine if the minimal polynomial splits into disctinct linear factors, $q_A(t) = (t-\lambda_1) \ldots (t-\lambda_k)$ where $\lambda_1, \ldots, \lambda_k$ are the distinct eigenvalues of the matrix A. In that case, the matrix A is diagonalizable and furthermore, we also know that $\lambda_1, \ldots, \lambda_k$ are all the possible distinct eigenvalues of the matrix A. If $\lambda_1, \ldots, \lambda_k$ are real, then A is real diagonalizable. In certain cases, the result of Theorem (4.2) can be applied immediately. For example, the class of idempotent matrices can easily be seen to be real diagonalizable. Since the polynomial $f(t) = t(t-1)$ is an annihilating polynomial for any idempotent matrix, where $A^2 = A$, by the observation above, $f(t)$ must be the minimal polynomial for any idempotent matrix. Moreover, an idempotent matrix A is real diagonalizable with exactly two distinct real eigenvalues, zero and one.

The method described above seems to be an easy way of determining diagonalizability of a matrix. The determination is purely based on whether there exists an annihilating polynomial that splits into distinct linear factors. In practice, one is still faced with computational difficulties in determining the minimal polynomial of a complex matrix unless it possesses some exploitable features, as in the case of an idempotent matrix. The next section develops a method that resolves this problem.

4.2 Determining the diagonalization of a matrix

We now present a systematic approach to determine if a matrix $A \in M_n(\mathbb{C})$ is real diagonalizable. Applying the MMP algorithm to a complex matrix gives the minimal polynomial of A. Once the minimal polynomial of the matrix A is obtained, determining whether the matrix A is real diagonalizable is a simple matter.

First, notice that if the eigenvalues of a matrix are all real, then the coefficients of the minimal polynomial must be all real. Secondly, if the minimal polynomial $q_A(t)$ has degree k, then it must have k distinct real roots for the matrix A to be real diagonalizable. To that end, we can use Sturm's sequence [11, Theorem 2.5.4] for finding the number of real roots of a real polynomial. In order to apply Sturm's theorem we will use Cauchy's bound [7, Exercise 27, pp. 316] to find an interval in which we will obtain number of the real distinct roots of the minimal polynomial.

We now formally state Cauchy's bound and Sturm's theorem.

Theorem 4.3 *[7, Exercise 27, pp. 316] Let* $p(x) = x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0,\ a_0 \neq 0$ *be a polynomial and* r *be any root of* $p(x) = 0$*, then*
$|r| \leq 1 + max\{|a_0|, |a_1|, \ldots, |a_{n-1}|\}.$

Let $f_0(x)$ be a polynomial. Denote the derivative of $f_0(x)$ by $f_1(x)$. Using the

Euclidean Algorithm calculate

$$\begin{aligned}
f_0(x) &= q_1(x)f_1(x) - f_2(x) \\
f_1(x) &= q_2(x)f_2(x) - f_3(x) \\
&\vdots \\
f_{k-2}(x) &= q_{k-1}(x)f_{k-1}(x) - f_k(x) \\
f_{k-1}(x) &= q_k(x)f_k(x)
\end{aligned}$$

where the degree of $f_i(x)$ is strictly lower than the degree of $f_{i-1}(x)$ for $1 \leq i \leq k$. The polynomial $f_k(x)$ divides $f_{k-1}(x)$, and $f_k(x)$ is the greatest common divisor of the original function $f_0(x)$ and its derivative $f_1(x)$. The sequence $f_0(x), f_1(x), \ldots f_k(x)$ is called the Sturm sequence for the polynomial $f_0(x)$.

We now state Sturm's theorem.

Theorem 4.4 *[11, Theorem 2.5.4] The number of distinct real zeros of a polynomial $f(x)$ with real coefficients in $[a, b]$ is equal to the difference of sign changes in $\{f_0(a), f_1(a), \ldots f_k(a)\}$ and $\{f_0(b), f_1(b), \ldots f_k(b)\}$.*

Example 4.1 In this example we determine whether the matrix

$$A = \begin{bmatrix} 1 & -1 & \frac{1}{2} & \frac{1}{2} \\ 1 & -2 & \frac{3}{2} & \frac{1}{2} \\ 1 & -2 & 1 & 1 \\ 18 & -3 & -4 & -1 \end{bmatrix}$$

is or is not real diagonalizable by using the systematic approach we have developed. First, using the MMP algorithm, we compute the minimal polynomial of the matrix A which results in $q_A(t) = t^4 + t^3 - 3t^2 - 4t - 4$. Since the coefficients of the minimal polynomial are real, we continue the process to determine whether the matrix A

is or is not diagonalizable. In order to find a suitable interval to apply Sturm's theorem, we use Theorem (4.3) to calculate $|r| \leq 1 + max\{1, 3, 4\} = 5$. Thus, a suitable interval to apply Sturm's theorem is $[-5, 5]$. Next we compute the Sturm sequence for the minimal polynomial $f_0(t) = q_A(t) = t^4 + t^3 - 3t^2 - 4t - 4$. The Sturm sequence is $f_0(t) = t^4 + t^3 - 3t^2 - 4t - 4,\ f_1(t) = 4t^3 + 3t^2 - 6t - 4,\ f_2(t) = \frac{27}{16}t^2 + \frac{21}{8}t + \frac{15}{4},\ f_3(t) = \frac{800}{81}t - \frac{256}{81},\ f_4(t) = -\frac{11907}{2500}$. Calculate $f_0(-5) = 441,\ f_1(-5) = -399,\ f_2(-5) = \frac{525}{16},\ f_3(-5) = \frac{-4256}{81},\ f_4(-5) = -\frac{11907}{2500}$. and $f_0(5) = 651,\ f_1(5) = 541,\ f_2(5) = \frac{945}{16},\ f_3(5) = \frac{416}{9},\ f_4(5) = -\frac{11907}{2500}$. To determine the number of real roots we calculate the number of sign changes in the values of $f_i(-5), i = 0, \ldots 4$ and $f_i(5), i = 0, \ldots 4$ and take the difference. The number of real roots is $3 - 1 = 2$ in this case. We conclude that the matrix A is not real diagonalizable, since the minimal polynomial has degree four and only two of the root are real.

If a matrix A is known to be real diagonalizable, then the degree of the minimal polynomial is the number of distinct eigenvalues of A. It is a simple matter to determine if a real diagonalizable matrix A has n distinct eigenvalues using our method. To see how this can be done, let $A \in M_n(\mathbb{C})$ be a diagonalizable matrix. Using Gaussian elimination, we may take the diagonalizable matrix and transform it to a lower Hessenberg form as seen in section (3.2) in Chapter 3. Once the lower Hessenberg form is obtained, we can immediately identify whether the matrix has n distinct eigenvalues. If the lower Hessenberg matrix is unreduced, then the minimal polynomial is equal to the characteristic polynomial and the matrix must have n distinct eigenvalues by Theorem (3.1). If the lower Hessenberg matrix is in reduced form, then the minimal polynomial is not equal to the characteristic polynomial and the matrix may have repeated eigenvalues.

Example 4.2 In this example we determine if the matrix

$$A = \begin{bmatrix} 2 & 1 & -2 & 3 \\ 1 & 1 & 4 & 0 \\ -2 & 4 & 0 & 1 \\ 3 & 0 & 1 & 1 \end{bmatrix} \in M_4(\mathbb{R})$$

has distinct eigenvalues. Since A is symmetric, it is real diagonalizable. Using the matrices $S_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & -3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $S_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-1}{7} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $S_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -22 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, and $S_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{-7}{55} \end{bmatrix}$, we reduce the matrix A to a lower Hessenberg by multiplying

$$(S_1S_2S_3S_4)^{-1}A(S_1S_2S_3S_4) = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 14 & -7 & 1 & 0 \\ 80 & -28 & \frac{34}{7} & 1 \\ \frac{-165}{7} & 0 & \frac{55}{49} & \frac{29}{7} \end{bmatrix} \in M_4(\mathbb{R}).$$

Since the lower Hessengerg matrix is unreduced, the matrix A has distinct eigenvalues by Theorem (3.1).

4.3 Pre-Hermitian Matrices

In the case of unitary similarity, the question of how to characterize unitary diagonalizability is completely settled. The following results are well known.

Theorem 4.5 *[7, Theorem 2.5.4] A matrix $A \in M_n(\mathbb{C})$ is unitarily diagonalizable if and only if A is normal.*

Theorem 4.6 *[7, Theorem 4.1.5] A matrix $A \in M_n(\mathbb{C})$ is unitarily real diagonalizable if and only if A is Hermitian.*

Different from diagonalization under similarity, diagonalization under unitary similarity comes down to determining whether a matrix A is normal, that is, if

$AA^* = A^*A$. It is even easier to determine whether a given matrix A is unitarily real diagonalizable. A matrix is unitarily real diagonalizable if $A^* = A$. The simplicity of identifying the class of unitarily real diagonalizable matrices suggests a new problem. Is there a class of matrices that can easily be identified as a real diagonalizable matrix?

In this section we study a class of matrices which is easily recognized to be real diagonalizable. We refer to them as the class of pre-Hermitian matrices. A pre-Hermitian matrix is defined to be diagonally similar to a Hermitian matrix, and therefore is real diagonalizable. We show that, like unitary similarity, there is a criterion that is easy to recognize in determining whether a matrix is pre-Hermitian. We also show that a Hermitian matrix is a type of pre-Hermitian matrix, and some properties of Hermitian matrices are carried to those that are pre-Hermitian. These include the interlacing property and eigenvalue majorization. We conclude this section by showing how we may use our algorithm to compute the minimal polynomial of a pre-Hermitian matrix. To that end, the following definition is needed.

Definition 4.1 *[9, p. 2498] A matrix $A \in M_n(\mathbb{C})$ is said to be **strongly sign-symmetric** if $a_{ii} \in \mathbb{R}$ for $i = 1, 2, \ldots, n$, $a_{ij}a_{ji} \geq 0$ for all $i, j = 1, 2, \ldots, n$, and $a_{ij}a_{ji} = 0$ only if $a_{ij} = a_{ji} = 0$. The set of all **strongly sign-symmetric** will be denoted by $P_s(\mathbb{C})$.*

In order to verify that a matrix is strongly sign-symmetric, we check that the Hadamard product, $A \circ A^T$, is non-negative. This is easy to do since we need only check that if $a_{ij} = |a_{ij}|e^{i\theta_{ij}}$, then $a_{ji} = |a_{ji}|e^{-i\theta_{ij}}$. It is easy to verify that all Hermitian matrices are strongly sign-symmetric; i.e., $H_n(\mathbb{C}) \subseteq P_s(\mathbb{C})$.

Unfortunately, matrices in $P_s(\mathbb{C})$ may not have all real eigenvalues. Consequently, a matrix in $P_s(\mathbb{C})$ may not be real diagonalizable. This is shown in the following example.

Example 4.3 Consider the matrix $A = \begin{bmatrix} 0 & 1 & 10 \\ 1 & 0 & 1 \\ .1 & 1 & 0 \end{bmatrix} \in M_3(\mathbb{C})$. Clearly, A is strongly sign-symmetric. We will show that the characteristic polynomial of A has only one real root. Since $p_A(\lambda) = det(A - \lambda I) = -\lambda^3 + 3\lambda + 10.1$, $p_A(\lambda)$ has a relative minimum at $(-1, 8.1)$, a relative maximum at $(1, 12.1)$, is decreasing on $(-\infty, -1)$ and $(1, \infty)$, and is increasing on $(-1, 1)$. Since $p_A(\lambda)$ is a cubic polynomial with negative leading coefficient, we see that $p_A(\lambda)$ has only one real root that is greater than 1 and two complex roots occurring as conjugates. Thus, $A \in P_s(\mathbb{C})$ while A has complex eigenvalues.

The following result motivates the class of pre-Hermitian matrices.

Proposition 4.1 *Let*

$$A = \begin{bmatrix} a_1 & b_1 & & & \mathit{0} \\ c_1 & a_2 & b_2 & & \\ & c_2 & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ \mathit{0} & & & c_{n-1} & a_n \end{bmatrix}$$

be a tridiagonal matrix. If $A \in P_s(\mathbb{C})$, then A is real diagonalizable. Moreover, if $b_i c_i > 0$ for $i = 1, \dots, n-1$, then the matrix A has distinct eigenvalues.

Proof. Suppose $A \in P_s(\mathbb{C})$ is tridiagonal, and let $X = diag(x_1, \dots, x_n) \in M_n(\mathbb{C})$ be an invertible diagonal matrix. We show that $x_i \neq 0$ can be chosen such that $X^{-1}AX$ is Hermitian which makes matrix A is real diagonalizable. Notice that

$$X^{-1}AX = \begin{bmatrix} a_1 & b_1\frac{x_2}{x_1} & & & 0 \\ c_1\frac{x_1}{x_2} & a_2 & b_2\frac{x_3}{x_2} & & \\ & c_2\frac{x_2}{x_3} & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-1}\frac{x_n}{x_{n-1}} \\ 0 & & & c_{n-1}\frac{x_{n-1}}{x_n} & a_n \end{bmatrix}.$$

If we can choose $x_i \neq 0$ such that $b_i\dfrac{x_{i+1}}{x_i} = \overline{c_i\dfrac{x_i}{x_{i+1}}}$, for $i = 1, \ldots, n$, then the product $X^{-1}AX$ is Hermitian.

Choose $x_1 = 1$, and for $i = 2, \ldots, n$ choose

$$x_i = \begin{cases} \sqrt{\dfrac{c_1c_2\ldots c_{i-1}}{b_1b_2\ldots b_{i-1}}} > 0 & b_{i-1} \neq 0 \\ \\ \sqrt{\dfrac{c_1c_2\ldots c_{i-2}}{b_1b_2\ldots b_{i-2}}} > 0 & b_{i-1} = 0 \end{cases}.$$

With this choice of $x_i's$ in X, it is routine to verify that A is similar to a Hermitian matrix. This shows that A is real diagonalizable. When $b_ic_i > 0$ for $i = 1, \ldots, n-1$, then the matrix A is an unreduced lower Hessenberg matrix. By Theorem (3.1), the minimal polynomial is equal to the characteristic polynomial of the matrix, and thus the eigenvalues of the matrix are distinct. $\square$

Proposition (4.1) shows how a non-Hermitian class of matrices can be determined to be real diagonalizable via a simple diagonal similarity. This observation motivates the next definition.

Definition 4.2 *A matrix $A \in P_s(\mathbb{C})$ is said to be* ***pre-Hermitian*** *if there exists $X \in D_n(\mathbb{C})$ such that $X^{-1}AX \in H_n(\mathbb{C})$. The class of pre-Hermitian matrices will be denoted by $PH_n(\mathbb{C})$.*

Diagonal simlarities have a long history and a variety of uses. For example, see [6], [4] and [10]. The class of pre-Hermitian matrices have been generalized to the class of matrices known as Hermitianable matrices [9]. In [9], the focus of study is on Schur type stability properties of Hermitainable matrices.

Here we focuc on the characteristics of pre-Hermitian matrices.

Proposition 4.2 $H_n(\mathbb{C}) \subset PH_n(\mathbb{C}) \subset RD_n(\mathbb{C})$

Proof. We first show that $H_n(\mathbb{C}) \subseteq PH_n(\mathbb{C})$. Let $A \in H_n(\mathbb{C})$ and take $X = I_n$. Then $X^{-1}AX = A \in PH_n(\mathbb{C})$, since A is Hermitian. So $H_n(\mathbb{C}) \subseteq PH_n(\mathbb{C})$. Next, let $A \in PH_n(\mathbb{C})$. Then there is an $X \in D_n(\mathbb{C})$ such that $X^{-1}AX = H$, where $H^* = H$. This means that, there is a unitary matrix U such that $U^*X^{-1}AXU = diag(\lambda_1, \ldots, \lambda_n)$, where λ_i are real numbers. Hence, $PH_n(\mathbb{C}) \subseteq RD_n(\mathbb{C})$. Next, we show that the inclusion is proper. Let $A = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \in M_2(\mathbb{R})$. Then A is not Hermitian. Choose $X = diag(2, \sqrt{2})$. Then $X^{-1}AX = \begin{bmatrix} 1 & \sqrt{2} \\ \sqrt{2} & 1 \end{bmatrix}$. This shows that $H_n(\mathbb{C}) \subset PH_n(\mathbb{C})$. Now let $A = \begin{bmatrix} 10 & 1 \\ 0 & 1 \end{bmatrix} \in M_2(\mathbb{R})$. Then the characteristic polynomial of A is $p_A(t) = (10-t)(1-t)$. Thus, A is diagonalizable. However there are no $X \in D_2(\mathbb{C})$ such that $X^{-1}AX$ is Hermitian, which shows $PH_n(\mathbb{C}) \subset RD_n(\mathbb{C})$ and the proof is complete. $\square$

Note that in the proof of Proposition (4.1) we showed that the diagonal matrix X can be chosen such that all entries on the diagonal could be taken to be positive. In general, this holds for pre-Hermitian matrices.

Proposition 4.3 *A matrix $A \in PH_n(\mathbb{C})$ if and only if there is a positive diagonal matrix $Y = diag(y_1, \ldots, y_n)$, with $y_i > 0$ for $i = 1, \ldots, n$ such that $(AY)^* = (AY)$.*

Proof. Let $A \in PH_n(\mathbb{C})$. Then there is an $X \in D_n(\mathbb{C})$ such that $X^{-1}AX = H \in H_n(\mathbb{C})$. Since H is a Hermitian matrix, $H = X^{-1}AX = X^*A^*(X^*)^{-1} =$

H^*. Hence $AXX^* = XX^*A^*$. Then $Y = XX^*$ is a positive diagonal matrix and $AY = YA^* = (AY)^*$, as desired. Conversely, let $(AY)^* = AY$ for some positive diagonal matrix $Y = diag(y_1, \ldots, y_n)$. Let $X = diag(\sqrt{y_1}, \ldots, \sqrt{y_n})$, the positive definite square root of Y. Then $Y = X^2 = XX^*$. Now $(AY)^* = (AY)$ gives $XX^*A^* = AXX^*$, and this yields $X^{-1}AX = X^*A^*(X^*)^{-1} = (X^{-1}AX)^*$. Thus $X^{-1}AX$ is Hermitian, or $A \in PH_n(\mathbb{C})$. $\square$

An equivalent statement to Proposition (4.3) is that $A \in PH_n(\mathbb{C})$ if and only if $A^* = X^{-1}AX$ for some positive diagonal matrix X. This observation is a useful tool to show when a matrix is pre-Hermitian. The following results are basic properties and characterizations of pre-Hermitian matrices.

Proposition 4.4 *Let $A \in PH_n(\mathbb{C})$. Then $A^n \in PH_n(\mathbb{C})$ for all n.*

Proof. Let $A \in PH_n(\mathbb{C})$. Then there is an $X \in D_n(\mathbb{C})$ such that $X^{-1}AX = H \in H_n(\mathbb{C})$. Thus $H^k = X^{-1}A^kX \in PH_n(\mathbb{C})$ for $k = 1, \ldots, n$, and the result follows. $\square$

Proposition 4.5 *A matrix $A \in PH_n(\mathbb{C})$ if and only if $V^{-1}AV = \Lambda$ where Λ is a real diagonal matrix and $V = X^{-1}UX$ for some $U \in U_n(\mathbb{C})$.*

Proof. Suppose $A \in PH_n(\mathbb{C})$. Then there exists an $X \in D_n(\mathbb{C})$ such that $X^{-1}AX = H \in H_n(\mathbb{C})$. Therefore there is a unitary $U \in U_n(\mathbb{C})$ such that $U^*X^{-1}AXU = U^*HU = \Lambda$, where Λ is a real diagonal matrix. Then $\Lambda = X^{-1}\Lambda X = X^{-1}U^*XAX^{-1}UX = V^{-1}AV$, where $V = X^{-1}UX$. Conversely, let $V^{-1}AV = \Lambda$, where Λ is a real diagonal matrix and $V = X^{-1}UX$ for some invertible diagonal matrix X, and $U \in U_n(\mathbb{C})$. Then $\Lambda = (X^{-1}UX)^{-1}A(X^{-1}UX) = X^{-1}U^*XAX^{-1}UX$. This yields $U\Lambda U^* = XAX^{-1}$, and since $U\Lambda U^*$ is Hermitian we know that $A \in PH_n(\mathbb{C})$. $\square$

Another class of matrices that are important in studying pre-Hermitian matrices is the cycle balanced matrix.

Definition 4.3 *[9, p. 2498] A matrix A* $M_n(\mathbb{C})$ *is said to be cycle balanced if for any sequence* $i_1, i_2, \ldots, i_k \in \{1, 2, \ldots, n\}$ *we have*

$$a_{i_1 i_2} a_{i_2 i_3} \cdots a_{i_k i_1} = a_{i_1 i_k} a_{i_k i_{k-1}} \cdots a_{i_2 i_1}.$$

The following theorem lists equivalent conditions for a matrix to be pre-Hermitian. The equivalence of (1) and (2) was shown in [13] and the remaining implications were shown above in Proposition (4.2) and the comments thereafter.

Theorem 4.7 *[9, p. 2498] For* $A \in M_n(\mathbb{C})$*, the following statements are equivalent:*

1. *A is pre-Hermitian.*

2. *A is cycle balanced and strongly sign-symmetric.*

3. *There exists a positive diagonal matrix K such that* $K^{-1}AK$ *is Hermitian.*

4. *There is a positive diagonal matrix Y such that* $(AY)^* = AY$*.*

We now turn our attention to properties that the class of pre-Hermitian matrices share with the class of Hermitian matrices. For this reason we give the following definitions and observations, beginning with the definition of a principal submatrix of a matrix.

Definition 4.4 *[7, Definition 0.7.1] Suppose* $A \in M_n(\mathbb{C})$*, and let the indexed set* $\alpha = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$*, where the* i_j*'s are listed in increasing order, be given. Then the principal submarix of A, denoted by* $A(\alpha)$*, is the matrix that lies in the rows and columns of A indexed by* α*.*

It is well known that the every principal submatrix of a Hermitian matrix is Hermitian [7, Exercise 1, pp. 174]. The class of pre-Hermitian matrices share this property with the Hermitian matrices. The result is our next proposition.

Proposition 4.6 *Let A be pre-Hermitian. Then every principal submatrix of A is pre-Hermitian.*

Proof. Suppose $A \in PH_n(\mathbb{C})$ and let $\alpha = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ be an indexed set such that $A(\alpha) \in M_k(\mathbb{C})$ is the principal submatrix of A indexed by α. We will show that $A(\alpha)$ is a pre-Hermitian matrix. Since $A \in PH_n(\mathbb{C})$, there is an invertible diagonal matrix $X = diag(x_1, \ldots, x_n) \in D_n(\mathbb{C})$ such that $X^{-1}AX = H$, where $H = H^*$. Let $H(\alpha) \in M_k(\mathbb{C})$ be the principal submatrix of H indexed by α. Then $H(\alpha)$ is a Hermitian matrix such that $H(\alpha) = (X^{-1}AX)(\alpha) = X^{-1}(\alpha)A(\alpha)X(\alpha)$, where $X(\alpha)$ is a k-by-k invertible diagonal matrix. We conclude the principal submatrix $A(\alpha)$ of A is pre-Hermitian. $\square$

One salient feature that can be observed from the proof of Proposition (4.6) is that if a pre-Hermitian matrix A is diagonally similar to a Hermitian matrix H, then for any indexed set α, the principal submatrix $A(\alpha)$ is precisely diagonally similar to the principal submatrix $H(\alpha)$ of the Hermitian matrix H. An important consequence of this observation is that the class of pre-Hermitian matrices maintains some desirable properties of the class of Hermitian matrices. The interlacing property of the eigenvalues between the principal submatrices is preserved, and the majorization of the vector of diagonal elements of the matrix and the vector of the eigenvalues is also preserved.

Next, Hermitian matrices have a nice spectral structure. Since principal submatrices of Hermitian matrices are Hermitian, and the eigenvalues of Hermitian matrices are real, it is natural to ask questions about the relationship between the eigenvalues of a Hermitian matrix and its principal submatrices. Moreover, the diagonal entries of Hermitian matrix are real. So it is natural to seek relationships between the diagonal entries and the real eigenvalues of a Hermitian matrix.

The well known interlacing property and the majorization of the eigenvalues of a Hermitian matrix by its diagonal entries characterize these properties. We define these terms to continue the discussion.

Definition 4.5 *Let $S_1 = \{\lambda_1, \ldots, \lambda_n\}$ and $S_2 = \left\{\lambda'_1, \ldots, \lambda'_{n+1}\right\}$ be two sets of real numbers that are listed in increasing order. Then S_1 and S_2 are said to interlace if*

$$\lambda'_1 \leq \lambda_1 \leq \lambda'_2 \leq \lambda_2 \leq \cdots \leq \lambda_{n-1} \leq \lambda'_n \leq \lambda_n \leq \lambda'_{n+1}.$$

Definition 4.6 *[7, Definition 4.3.24] Let $S_1 = \{\alpha_1, \ldots, \alpha_n\}$ and $S_2 = \{\beta_1, \ldots, \beta_n\}$ be sets of real numbers such that the α_i's and β_i are listed in increasing order. Then the set S_2 is said to majorize the set S_1 if*

$$\sum_{i=1}^{k} \beta_i \geq \sum_{i=1}^{k} \alpha_i, \quad k = 1, \ldots, n-1,$$

and

$$\sum_{i=1}^{n} \beta_i = \sum_{i=1}^{n} \alpha_i.$$

Definition 4.7 *Let $A \in M_n(\mathbb{C})$ be such that $A = \begin{bmatrix} A' & y \\ z^T & a_{nn} \end{bmatrix}$ with $A' \in M_{n-1}(\mathbb{C})$, $y, z \in \mathbb{C}^{n-1}$, and $a_{nn} \in \mathbb{R}$. Let $\sigma(A) = \{\lambda_1, \ldots, \lambda_n\}$, and $\sigma(A') = \left\{\lambda'_1, \ldots, \lambda'_{n-1}\right\}$ be the eigenvalues, listed in increasing order, of the matrices A and A', respectively. Assume that $\sigma(A)$ and $\sigma(A')$ are sets of real numbers. Then the matrices A and A' are said to have the interlacing property if $\sigma(A)$ and $\sigma(A')$ interlace.*

The statements of the interlacing and majorization properties are now given. The first relates the eigenvalues of a Hermitian matrix and its principal submatrices. The second relates the diagonal entries of a Hermitian matrix and its eigenvalues.

Theorem 4.8 *[7, Theorem 4.3.8] Let $A \in H_n(\mathbb{C})$, $y \in \mathbb{C}^n$ be a given vector, and $a \in \mathbb{R}$ be a given real number. Let $A' \in M_{n+1}(\mathbb{C})$ be the Hermitian matrix obtained by bordering A with y and a; i.e.,*

$$A' \equiv \begin{bmatrix} A & y \\ y^* & a \end{bmatrix}.$$

Let the eigenvalues of A and A' be denoted by $\{\lambda_1, \cdots, \lambda_n\}$ and $\{\lambda'_1, \cdots, \lambda'_{n+1}\}$, respectively, and assume that they are arranged in increasing order. Then these two sets of real eigenvalues interlace.

Theorem 4.9 *[7, Theorem 4.3.26] Let $A \in H_n(\mathbb{C})$. The set of diagonal entries of A majorizes the set of eigenvalues of A.*

Analogous results to Theorems (4.8) and (4.9) hold for the class of pre-Hermitian matrices.

Theorem 4.10 *Let $A \in PH_n(\mathbb{C})$. Then the matrix A has the interlacing property.*

Proof. Let $A \in PH_n(\mathbb{C})$, and write $A = \begin{bmatrix} A' & y \\ z^T & a_{nn} \end{bmatrix}$, where A' is the $(n-1)$-by-$(n-1)$ principal submatrix of A. Let $\sigma(A) = \{\lambda_1, \ldots, \lambda_n\}$, and $\sigma(A') = \{\lambda'_1, \ldots, \lambda'_{n-1}\}$ be the eigenvalues, listed in increasing order, of the matrices A and A' respectively. Since $A \in PH_n(\mathbb{C})$, $A' \in PH_{n-1}(\mathbb{C})$ by Proposition (4.6). Furthermore, there is a diagonal matrix $X = diag(x_1, \ldots, x_{n-1}, x_n) = \begin{bmatrix} x' & 0 \\ 0 & x_n \end{bmatrix} \in D_n(\mathbb{C})$ such that $X^{-1}AX = \begin{bmatrix} H' & y \\ y^* & a_{nn} \end{bmatrix} = H \in H_n(\mathbb{C})$, where is $H' = (X')^{-1}A'X'$ is the principal submatrix of the Hermitian matrix H. Since A is similar to H and A' is similar to H', we have $\sigma(A) = \sigma(H)$ and $\sigma(A') = \sigma(H')$. We conclude that A has the interlacing property by Theorem (4.8). □

Theorem 4.11 *Let $A \in PH_n(\mathbb{C})$. Then the set of diagonal entries of A majorizes the set of eigenvalues of A.*

Proof. Let A be diagonally similar to a Hermitian matrix H. Then A has the same diagonal entries and spectrum as the Hermitian matrix H. By Theorem (4.9), the result holds. □

Matrix recognition is an important question in matrix theory. Given a matrix $A \in M_n(\mathbb{C})$, it is important to determine what kind of matrix the matrix A is. In other words, it is important to identify the class of matrices in which A belongs. For special classes of matrices this is a simple task. For example, Hermitian matrices are easy to recognize, since we just check that $a_{ij} = \overline{a_{ji}}$ for all i and j. A few other classes of matrices that are easy to recognize are normal matrices, idempotent matrices, and unitary matrices. However, there are matrices that are very difficult to recognize. One such class is that of the totally positive matrices (see [3]). A matrix is totally nonnegative if each of its minors is nonnegative. This is a very difficult requirement to check in practice. Having a simple criterion that guarantees a matrix is in a certain class is a desirable property to have.

The following result gives an easy to recognize criterion to determine if a given matrix is pre-Hermitian. For the simplicity of the proof of the main theorem, it is assumed that the entries of the last row of the pre-Hermitian matrix are non-zero.

Proposition 4.7 *Let $A = [a_{ij}] \in P_s(\mathbb{C})$ such that $a_{nj} \neq 0$ for $j = 1, \ldots, n$. Then $A \in PH_n(\mathbb{C})$ if and only if $a_{ij}\overline{a_{ni}}a_{jn} = \overline{a_{ji}}a_{in}\overline{a_{nj}}$, for all $i, j = 1, \ldots, n$.*

Proof. Let $A \in PH_n(\mathbb{C})$. Then by proposition (4.3) there is a positive diagonal matrix $Y = diag(y_1, \ldots, y_n)$ with $y_i > 0$ for $i = 1, \ldots, n$ such that $AY = (AY)^*$. In matrix form, this means

$$\begin{bmatrix} a_{11}y_1 & a_{12}y_2 & \cdots & a_{1n}y_n \\ a_{21}y_1 & a_{22}y_2 & \cdots & a_{2n}y_n \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1}y_1 & a_{n2}y_2 & \cdots & a_{nn}y_n \end{bmatrix} = \begin{bmatrix} \overline{a_{11}}y_1 & \overline{a_{21}}y_1 & \cdots & \overline{a_{n1}}y_1 \\ \overline{a_{12}}y_2 & \overline{a_{22}}y_2 & \cdots & \overline{a_{n2}}y_2 \\ \vdots & \vdots & \vdots & \vdots \\ \overline{a_{1n}}y_n & \overline{a_{n2}}y_n & \cdots & \overline{a_{nn}}y_n \end{bmatrix}.$$

Equating the last column of the two matrices, we obtain $y_i = \dfrac{a_{in}}{\overline{a_{ni}}}y_n$, for $i = 1, \ldots, n$. Since $a_{ij}y_j = \overline{a_{ji}}y_i$, we have $a_{ij}\dfrac{a_{jn}}{\overline{a_{nj}}} = \overline{a_{ji}}\dfrac{a_{in}}{\overline{a_{ni}}}$. Thus, we get $a_{ij}\overline{a_{ni}}a_{jn} = \overline{a_{ji}}a_{in}\overline{a_{nj}}$ for all $i, j = 1, \ldots, n$. Conversely, assume $a_{ij}\overline{a_{ni}}a_{jn} = \overline{a_{ji}}a_{in}\overline{a_{nj}}$ for all $i, j = 1, \ldots, n$. Define $y_1 = \dfrac{a_{1n}}{\overline{a_{n1}}} > 0, y_2 = \dfrac{a_{2n}}{\overline{a_{n2}}} > 0, \ldots, \dfrac{a_{(n-1)n}}{\overline{a_{n(n-1)}}} > 0$, and $y_n = 1$. Let $Y = diag(y_1, \ldots, y_n)$. Then Y is a positive diagonal matrix such that $AY = [a_{ij}y_j] = [a_{ij}\dfrac{a_{jn}}{\overline{a_{nj}}}] = [\overline{a_{ji}}\dfrac{a_{in}}{\overline{a_{ni}}}] = [\overline{a_{ji}}y_i] = (AY)^*$. We conclude that $A \in PH_n(\mathbb{C})$ by Proposition (4.3). $\square$

In Proposition (4.7) we see that there are n^2 products to check to determine if a matrix $A \in M_n(\mathbb{C})$ is pre-Hermitian. However, we can significantly reduce the number of products we need to compute. For the discussion that follows, we assume that $a_{ij} \in \mathbb{R}$ for $i, j = 1, \ldots, n$. First, if $j = n$, then the triple product in Proposition (4.7) is always an identity, and it may be discarded from the n^2 products. Secondly, since we are finding a diagonal matrix $X = diag(x_1, \ldots, x_n)$ such that $X^{-1}AX$ is Hermitian, we are choosing x_i's where $i = 1, \ldots, n$ so that $a_{ij}\dfrac{x_j}{x_i} = a_{ji}\dfrac{x_i}{x_j}$. So we may also deduct the products in the case that $j = i$. Lastly, a_{ij} gives the same triple product as a_{ji} for $j \neq i$, which means it is unnecessary to compute the products of a_{ji} in Proposition (4.7) whenever $j < i$. Another way of saying this is that we only need to compute the triple product using a_{ij} for $i < j$, with $j = i + 1, \ldots, n - 1$. Putting this all together we obtain the number of triple products actually needed to check if a matrix is pre-Hermitian is $\sum_{i=0}^{n-2} i = \frac{(n-1)(n-2)}{2}$.

Remark 4.1 *In Proposition (4.7) the number of triple products required to check if a matrix is pre-Hermitian is* $\dfrac{(n-1)(n-2)}{2}$*. Moreover, the* $\dfrac{(n-1)(n-2)}{2}$ *triple products needed to compute are those involving* a_{ij} *where* $i < j$ *and* $j = i+1, \ldots, n-1$*.*

Remark (4.1) greatly reduces the amount of computation required to implement Proposition (4.7). For example, For matrices in $M_3(\mathbb{C})$ we see that there is only one triple product computation to determine whether the given matrix is pre-Hermitian. For matrices in $M_4(\mathbb{C})$ we see that there are only three triple products computed to determine whether the matrix is pre-Hermitian. The next example shows that the determination of whether a matrix is pre-Hermitian is indeed a computationally simple task.

Example 4.4 Let

$$A = \begin{bmatrix} -1 & -2 & 20 & 10 \\ -1 & 1 & -5 & 5 \\ 2 & -1 & 1 & 1 \\ 3 & 3 & 3 & -1 \end{bmatrix} \in M_4(\mathbb{R})$$

To determine if the matrix A is pre-Hermitian, it suffices to verify only three triple products in Proposition (4.7). Those three triple products are the triple products involving the entries a_{12}, a_{13}, and a_{23}. Accordingly, we must show that $a_{12}a_{41}a_{24} = a_{21}a_{14}a_{42}$, $a_{13}a_{41}a_{34} = a_{31}a_{14}a_{43}$, and $a_{23}a_{42}a_{34} = a_{32}a_{24}a_{43}$. In this example we have $-2 \times 3 \times 5 = -1 \times 10 \times 3$, $20 \times 3 \times 1 = 2 \times 10 \times 3$, and $-5 \times 3 \times 1 = -1 \times 5 \times 3$, respectively. Thus, by Proposition (4.7), the matrix A is pre-Hermitian.

This example shows the power of Proposition (4.7). To see if 4×4 matrix with complex entries has the interlacing property, majorization of eigenvalues by diagonal entries, and is real diagonalizable, we only had to verify three triple products.

The following result further characterizes the pre-Hermitian matrices.

Proposition 4.8 *Let $A \in M_n(\mathbb{C})$. Then A is real diagonalizable if and only if A is unitarily similar to a pre-Hermitian matrix.*

Proof. Suppose that $R^{-1}AR = \Lambda$, where Λ is a real diagonal matrix. Use the singular value decomposition of R to write $R = V^*\Sigma U$, where U and V are unitary matrices and Σ is a positive diagonal matrix. Then $R^{-1}AR = U^*\Sigma^{-1}VAV^*\Sigma U = \Lambda$. Multiply both sides of the equation on the left by Σ and on the right by Σ^{-1} to obtain $\Sigma U^*\Sigma^{-1}VAV^*\Sigma U\Sigma^{-1} = \Sigma\Lambda\Sigma^{-1} = \Lambda$. This shows that $VAV^* \in PH_n(\mathbb{C})$ by Proposition (4.5).

Now suppose A is unitarily similar to a pre-Hermitian matrix. In other words, $U^*AU \in PH_n(\mathbb{C})$. This means there exists $X \in D_n(\mathbb{C})$ such that $X^{-1}U^*AUX = H$, where $H = H^*$. Since $X^{-1}U^*AUX$ is Hermitian, there is a unitary V such that $V^*(X^{-1}U^*AUX)V = \Lambda$, where $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$ with $\lambda_i \in \mathbb{R}$ for $i = 1, \ldots, n$. If we set $R = UXV$, then $R^{-1}AR = \Lambda$, and we conclude that A is real diagonalizable. $\square$

It is natural to ask what type of similarities will preserve the pre-Hermitian structure. The following two results show the property of being pre-Hermitian is preserved under permutation and diagonal similarity.

Proposition 4.9 *$A \in M_n(\mathbb{C})$ is pre-Hermitian if and only if $\Pi^T A\Pi \in PH_n(\mathbb{C})$ for all permutation matrices $\Pi \in M_n(\mathbb{C})$.*

Proof. Assume that $\Pi^T A\Pi \in PH_n(\mathbb{C})$ for all permutation matrices Π. Since the identity matrix I_n is a permutation matrix $A = I_n^T AI_n \in PH_n(\mathbb{C})$. Now let $A \in PH_n(\mathbb{C})$. We show that $\Pi^T A\Pi \in PH_n(\mathbb{C})$ for any permutation matrices $\Pi \in M_n(\mathbb{C})$. Since A is pre-Hermitian, there is a $Y \in D_n(\mathbb{C})$ such that $Y^{-1}AY$ is Hermitian. We note for any permutation $\Pi \in M_n(\mathbb{C})$ there is a matrix $X \in D_n(\mathbb{C})$

such that $\Pi^T Y \Pi = X \in D_n(\mathbb{C})$; that is, $Y\Pi = \Pi X$. Thus for any permutation matrix $\Pi \in M_n(\mathbb{C})$ we have $\Pi^T Y^{-1} A Y \Pi = X^{-1} \Pi^T A \Pi X \in H_n(\mathbb{C})$, for some $X \in D_n(\mathbb{C})$. This shows that $\Pi^T A \Pi \in PH_n(\mathbb{C})$. □

Proposition 4.10 *Let $A \in PH_n(\mathbb{C})$. Then $D^{-1}AD \in PH_n(\mathbb{C})$ for all invertible diagonal matrices $D \in D_n(\mathbb{C})$.*

Proof. Let $A \in PH_n(\mathbb{C})$. Then there is a positive diagonal matrix Y_1 such that $Y_1^{-1} A Y_1 = A^*$ by Proposition (4.3). Suppose $B = D^{-1}AD$ for some $D \in D_n(\mathbb{C})$. We will show $B \in PH_n(\mathbb{C})$. Let $Y = D^{-1} Y_1 D^{-*}$. Then Y is a positive diagonal matrix such that $Y^{-1}BY = D^* Y_1^{-1} D B D^{-1} Y_1 D^{-*} = D^* Y_1^{-1} D D^{-1} A D D^{-1} Y_1 D^{-*} = D^* A^* D^{-*} = B^*$. Thus, by Proposition (4.3), $B \in PH_n(\mathbb{C})$. □

Combining the previous two propositions we obtain the following result about pre-Hermitian matrices.

Proposition 4.11 *Let $A \in PH_n(\mathbb{C})$. Then $\Pi^T X^{-1} A X \Pi$ and $X^{-1} \Pi^T A \Pi X \in PH_n(\mathbb{C})$ for $X \in D_n(\mathbb{C})$ and $\Pi \in \Pi_n$.*

It is well known that if a Hermitian matrix A is similar to a diagonal matrix, then the similarity can be taken as a unitary similarity. That is, let $A \in M_n(\mathbb{C})$ be Hermitian. If $\Lambda = R^{-1}AR$ for some invertible $R \in M_n(\mathbb{C})$ and $\Lambda \in D_n(\mathbb{R})$, then there is a unitary $U \in U_n(\mathbb{C})$ such that $\Lambda = R^{-1}AR = U^*AU$. An analogous result holds for pre-Hermitian matrices. For the purpose of stating our next proposition, a definition, theorem, and corallary are needed.

Definition 4.8 *An invertible matrix $V \in M_n(\mathbb{C})$ is said to be* ***pre-unitary*** *if $V = X^{-1}UX$, for some diagonal $X \in D_n(\mathbb{C})$ and $U \in U_n(\mathbb{C})$.*

Theorem 4.12 *[7, Exercise 9, pp. 97] Let $A \in M_n(\mathbb{C})$, $B \in M_m(\mathbb{C})$, and $C \in M_{n,m}(\mathbb{C})$ be given. Then the matrix equation $AX - XB = C$ has a unique solution if and only if $\sigma(A) \cap \sigma(B) = \emptyset$.*

Corollary 4.1 *[15, Exercise 4.31, pp. 83] Let $A \in M_n(\mathbb{C})$ and suppose $P \in M_n(\mathbb{C})$ is positive definite. If $AP = PA$, then $AP^{+\frac{1}{2}} = P^{+\frac{1}{2}}A$, where $P^{+\frac{1}{2}}$ is the positive definite square root of the positive definite matrix $P \in M_n(\mathbb{C})$.*

Proposition 4.12 *Let $A \in PH_n(\mathbb{C})$. If $R^{-1}AR = \Sigma$, where $\Sigma \in D_n(\mathbb{R})$, then there is a pre-unitary matrix $V \in M_n(\mathbb{C})$ such that $\Sigma = R^{-1}AR = V^{-1}AV$.*

Proof. Let $A \in PH_n(\mathbb{C})$. Suppose $A = R^{-1}\Sigma R$ for some invertible matrix $R \in M_n(\mathbb{C})$ and a positive diagonal matrix $\Sigma \in D_n(\mathbb{R})$. Then $A^* = R^*\Sigma R^{-*}$, and by Proposition (4.3) there is a positive diagonal matrix $Y \in D_n(\mathbb{R})$ such that $A^* = Y^{-1}AY$. Hence, we have $Y^{-1}AY = Y^{-1}R^{-1}\Sigma RY = R^*\Sigma R^{-*} = A^*$, or $\Sigma RYR^* = RYR^*\Sigma$. Set $B = RYR^*$. Then B is a positive definite matrix. By Corollary (4.1), $\Sigma B^{+\frac{1}{2}} = B^{+\frac{1}{2}}\Sigma$. Notice that $I = (B^{+\frac{1}{2}})^{-1}B(B^{+\frac{1}{2}})^{-1} = (B^{+\frac{1}{2}})^{-1}RYR^*(B^{+\frac{1}{2}})^{-1} = (B^{+\frac{1}{2}})^{-1}RY^{+\frac{1}{2}}Y^{+\frac{1}{2}}R^*(B^{+\frac{1}{2}})^{-1} = ((B^{+\frac{1}{2}})^{-1}RY^{+\frac{1}{2}})((B^{+\frac{1}{2}})^{-1}RY^{+\frac{1}{2}})^*$. In other words, $(B^{+\frac{1}{2}})^{-1}RY^{+\frac{1}{2}}$ is a unitary matrix. Set $V = (B^{+\frac{1}{2}})^{-1}RY^{+\frac{1}{2}} \in U_n(\mathbb{C})$. Now $A = R^{-1}\Sigma R = R^{-1}B^{+\frac{1}{2}}\Sigma(B^{+\frac{1}{2}})^{-1}R = Y^{+\frac{1}{2}}(Y^{+\frac{1}{2}})^{-1}R^{-1}B^{+\frac{1}{2}}\Sigma(B^{+\frac{1}{2}})^{-1}RY^{+\frac{1}{2}}(Y^{+\frac{1}{2}})^{-1} = Y^{+\frac{1}{2}}V^{-1}(Y^{+\frac{1}{2}})^{-1}\Sigma Y^{+\frac{1}{2}}V(Y^{+\frac{1}{2}})^{-1}$, where $Y^{+\frac{1}{2}}V(Y^{+\frac{1}{2}})^{-1}$ is a pre-unitary matrix and the proof is complete. □

We now turn our attention to computing the minimal polynomial of pre-Hermitian matrices. The idea behind computing the minimal polynomial of pre-Hermitian matrices is simple: Pre-Hermitian matrices are defined to be matrices that are diagonally similar to a Hermitian matrix. Thus, if a matrix A is pre-Hermitian, we may apply the diagonal similarity to obtain $X^{-1}AX = H$, where H is a Hermitian matrix. Once the pre-Hermitian matrix has been transformed to a Hermitian matrix, finding the minimal polynomial can be achieved by transforming the Hermitian matrix to a tridiagonal matrix as discussed in Section (3.2). Moreover, using Remark (4.1), we may easily check if a given matrix $A \in M_n(\mathbb{C})$ is pre-Hermitian. We illustrate this in the next example.

Example 4.5 Let

$$A = \begin{bmatrix} 1 & 4 & -8 \\ 1 & 10 & 3 \\ -2 & 3 & 1 \end{bmatrix} \in M_3(\mathbb{R}).$$

Applying the observation made in Remark (4.1), we only need to check that $a_{12}a_{31}a_{23} = a_{21}a_{13}a_{23}$. This computation is $4 \times (-2) \times 3 = 1 \times (-8) \times 3$, so A is pre-Hermitian. Choosing $X = diag(2, 1, 1)$, we calculate

$$X^{-1}AX = \begin{bmatrix} 1 & 2 & -4 \\ 2 & 10 & 3 \\ -4 & 3 & 1 \end{bmatrix}.$$

Now that we have found the diagonal matrix X such that $X^{-1}AX$ is Hermitian, we use a Givens matrix to tridiagonalize this Hermitian matrix. Choosing

$$U(\theta, 1, 3) = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

Next, we calculate

$$U^*(\theta, 1, 3)X^{-1}AXU(\theta, 1, 3) = \begin{bmatrix} 5 & -\frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & 10 & \frac{5}{\sqrt{2}} \\ 0 & \frac{5}{\sqrt{2}} & -3 \end{bmatrix}.$$

Since the matrix A has been tridiagonalized, we may use Remark (3.1) to compute the minimal polynomial of A. We start with $p_0(t) = 1$ and $p_1(t) = t - 10$.

Then

$$\begin{aligned} p_2(t) &= (t - a_2)p_1(t) - b_1c_1p_0(t) \\ &= t^2 - 15t + \frac{99}{2} \end{aligned}$$

and

$$\begin{aligned} p_3(t) &= (t - a_3)p_2(t) - b_2c_2p_1(t) \\ &= t^3 - 12t^2 - 8t + 211 \end{aligned}$$

The minimal polynomial of matrix A is $q_A(t) = t^3 - 12t^2 - 8t + 211$.

CHAPTER 5

Summary and topics of further study

In this dissertation, we introduced the MP algorithm which gives us a new and simple way to calculate the minimal polynomial of any matrix $A \in M_n(\mathbb{C})$. We showed that the MP algorithm is extremely easy to implement to matrices of relatively small size. We then gave a modified version of the MP algorithm that is easier to apply. After we established the MMP algorithm, we gave several applications of the minimal polynomial.

After we considered the applications of the MMP algorithm, we showed that the MMP algorithm significantly simplifies when it is applied to matrices of special structure. The two special strucutred matrices we considered were the class of lower Hessenberg matrices and the class of tridiagonal matrices. In the case of an unreduced lower Hessenberg matrix, we saw that only the first row of the matrix was necessary in the MMP alogrithm to compute its minimal polynomial. For the tridiagonal matrix, we gave a recursive formula to compute its minimal polynoimial.

We concluded this dissertation by giving a systematic way for determining if a matrix is or is not real diagonalzable. We also defined a new class of real diagonalizable matrices, the pre-Hermitian matrices. We showed that there is an computationally simple way to verify if a matrix is pre-Hermitian and showed how we may compute the minimal polynomial of a pre-Hermitian matrix.

Topics of future study include:

- We saw that the *rvec* of matrix was a useful operation in the MP algorithm to compute the minimal polynomial of a matrix. Further, for the class of lower Hessenberg matrices we were able to modify the *rvec* operation and use only the first row of the matrix to compute the minimal polynomial. It is natural to seek out other types of *rvec*-type operations for other classes of matrices.

- We would like to explore the MMP algorithm on different types of matrices. For example, the class of $(0, 1)$-matrices are defined to be matrices that only have 0's and 1's for its entries. Given this structure, the MMP algorithm will be very simple to implement.

- We would like to find other applications of the MMP algorithm and the GU matrix.

- Computer implemention and numerical considerations of the MMP algorithm.

- We continue our study of pre-Hermitian matrices. As it is shown in section (4.3), we have the following containments $H_n(\mathbb{C}) \subset PH_n(\mathbb{C}) \subset RD_n(\mathbb{C})$. We would like to extend this containment; that is, are there any other classes of matrices $C_1, \cdots, C_i$ such that $H_n(\mathbb{C}) \subset C_1 \subset \cdots \subset C_i \subset PH_n(\mathbb{C}) \subset RD_n(\mathbb{C})$? If any such class of matrices exist, we would like to study their intrinsic properties.

- We would like to explore other types of class of pre-matrices and study their basic properties. So far we have only defined pre-Hermitian matrices. We can similarly define pre-unitary, pre-normal, pre-positive definite, etc. For example, a matrix is positive definite if and only if the determinant of every leading principal submatrix is positive [7, Theorem 7.2.5]. Does something similar hold for pre-positive definite?

REFERENCES

[1] S. Bialas and m. Bialas. An algorithm for the calculation of the minimal polynomial. *Bulletin of the Polish Academy of sciences*, 56, 2008.

[2] Charles W. Curtis. *Linear Algebra: An introductory approach.* Spinger, New York, 1984.

[3] Shaun M. Fallat and Charles R. Johnson. *Totally Nonnegative Matrices.* Princeton Series in Applied Mathematics, Princeton University Press, 2011.

[4] Xin-Lei Fenga, Zhongshan Li, and Ting-Zhu Huang. Is every nonsingular matrix diagonally equivalent to a matrix with all distinct eigenvalues? *Linear Algebra and its Applications*, 436:120125, 2012.

[5] F. R. Gantmacher. *The theory of matrices volume 1.* Chelsea Publishing company, New York, 1959.

[6] Daniel Hershkowitz and Hans Schneider. One-sided simultaneous inequalities and sandwich theorems for diagonal similarity and diagonal equivalence of nonnegative matrices. *Electronic Journal of Linear Algebra*, 10:81–101, 2003.

[7] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, New York, 1985.

[8] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis.* Cambridge University Press, New York, 1991.

[9] Charles R. Johnson, Terry D. Lenker, and Sivaram K. Narayan. Schur-type stability properties and complex diagonal scaling. *Linear Algebra And Its Applications*, 429 (2008) 2497-2520, 2008.

[10] Charles R. Johnson, David P. Stanford, D. Dale Oleskyt, and P. van den Driessches. Dominant eigenvalues under trace-preserving diagonal perturbations. *Linear Algebra and its Applications*, 212213:415–435, 1994.

[11] B. E. Meserve. *Fundamental concepts of algebra.* Dover Publications, Mineola, New York, 1982.

[12] Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra.* SIAM, Philadelphia, 2000.

[13] P. Nowosad. Symmetrization of matrices by diagonal matrices. *MRC Technical Summary report 1083. Mathematics Research Center, University of Wisconsin, Madison, Wisc., 1970*, 1970.

[14] Hans Schneider and George P. Barker. *Matrices and Linear Algebra.* Dover, New York, 1968.

[15] Fuzhen Zhang. *Linear Algebra: Challenging problems for students, second edition.* The Johns Hopkins University Press, Baltimore, 2009.