18.821 SPRING 2008: GENERATING MATRICES

JUSTIN CURRY, MICHAEL FORBES, MATTHEW GORDON

ABSTRACT. In this paper we analyze the problem of when the monomials of two $n \times n$ matrices X and Y generate the space M of all $n \times n$ matrices. Theoretical lower and upper bounds on the maximum required degree are established for matrices over any field. Specific examples of sufficient conditions on X and Y to generate M are provided. The complex case is considered to show that these sufficient conditions can be extended to a condition satisfied on a dense open subset of $M = \mathbb{C}_{n\times n}$. Several concrete examples in the real case are calculated using computational means and the growth of independent monomials at each degree is illustrated. Probabilistic methods are then used to explore a wide class of matrices that generate M with different efficiencies and this analysis suggests our asymptotic bounds are not tight.

1. INTRODUCTION

Certain sets of real $n \times n$ matrices M can be used to form a basis for the space of real $n \times n$ matrices. For example, the n^2 elementary matrices E_{ij}, which have a 1 in the ij^{th} place and zeros everywhere else, are the standard basis. A more interesting example of a basis might be formed from the pair of matrices X,Y using a subset of the monomials of X and Y:

$$\{I, X, Y, XX, XY, YX, YY, XXX, XXY, XYX, XYY, YXX, YXY, \ldots\}$$

where I is the identity matrix. If some subset of this set spans M we say that X and Y *generate* M. In this paper, we examine which X,Y pairs generate M.

Definition 1.1. For $n \times n$ real matrices X and Y, the **set of degree d monomials** is the set of terms $\{Z_1 \cdots Z_d\}$ where $Z_i \in \{X, Y\}$ and $d \geq 0$. The set of monomials of degree 0 corresponds to the empty matrix product, which is the $n \times n$ identity matrix.

It is clear from this definition that there are 2^d monomials of degree d. Furthermore, the set of monomials for the pair X,Y is the union of the monomials of degree $d \geq 0$. This is simply the set of words over the alphabet $\{X, Y\}$ with empty word I.

Definition 1.2. Let M be the n^2-dimensional vector space of $n \times n$ real matrices. A pair of real matrices X,Y **generates** M if the set of monomials spans M. Gen$\{X, Y\}$ denotes the span of the monomials.

The question of whether an arbitrary set of real matrices generates M could also be explored, but this paper focuses on sets of size two. As Proposition 1.3 shows, sets of size one cannot generate M and sets of size larger than two are more difficult to work with.

Proposition 1.3. *No $n \times n$ matrix X generates M for any field.*

Proof. Let p be the characteristic polynomial of the $n \times n$ matrix X. $p(X)$ is equal to the zero matrix by the Cayley-Hamilton Theorem, so there must be a linear relation of the form:

$$a_n X^n + a_{n-1} X^{n-1} + \cdots + a_0 I = 0_{n \times n}$$

Accordingly, X^n is a linear combination of $I, X, \ldots, X^{n-1}$, so $X^n \in \text{Span}\{I, X, \ldots, X^{n-1}\}$. We can then use matrix multiplication to deduce that $X^{n+i} \in \text{Span}\{X^i, X^{i+1}, \ldots, X^{n+i-1}\}$. Applying this fact inductively gives that $\text{Span}\{X^i, X^{i+1}, \ldots, X^{n+i-1}\} \subseteq \text{Span}\{I, X, \ldots, X^{n-1}\}$ and so $X^{n+i} \in \text{Span}\{I, X, \ldots, X^{n-1}\}$ for all $i \geq 0$. Therefore, $\text{Gen}\{X\} = \text{Span}\{I, X, \ldots, X^{n-1}\}$ and thus has dimension at most n. As M has dimension n^2, $\text{Gen}\{X\} \subset M$ and X thus fails to generate M.[1] □

Thus, a set of at least two real matrices is necessary but not sufficient for generating M. For example, two identical real matrices cannot generate M as a simple consequence of Proposition 1.3.

It is useful to be able to talk about how efficiently a pair of monomials generates the space of real $n \times n$ matrices. In order to be able to do this, we introduce a notion of efficiency that we work with in this paper.

Definition 1.4. For any pair of $n \times n$ matrices X,Y which generate M, let $d(X, Y) = d$ denote the smallest integer such that the monomials of degree $\leq d$ span M.

We can now define the lower bound and the upper bound for generating M with respect to d. We are naturally interested in these because they give us asymptotic bounds on the efficiency with which M can be generated.

Definition 1.5. Let δ_n be the smallest integer such that there exists a generating pair X,Y of $n \times n$ matrices with $d(X, Y) = \delta_n$. We call δ_n the **lower bound** for generating the n^2-dimensional vector space of $n \times n$ real matrices.

Definition 1.6. Let Δ_n be the largest integer such that there exists a generating pair X,Y of $n \times n$ matrices with $d(X, Y) = \Delta_n$. We call Δ_n the **upper bound** for generating the n^2-dimensional vector space of $n \times n$ real matrices.

We also want to analyze how quickly two matrices span M, so we define the growth function for a pair of real matrices.

Definition 1.7. For a pair of $n \times n$ matrices X and Y, let V_i denote the subspace of M spanned by monomials in X and Y of degree at most i. The **growth function** for the pair X,Y is $g(i) = \dim V_i - \dim V_{i-1}$.

The following proposition captures an important property of $\text{Gen}\{X, Y\}$ that will be useful for the rest of the paper.

Proposition 1.8. *If* $\text{Gen}\{X, Y\} = M$, *then* $\text{Gen}\{P^{-1}XP, P^{-1}YP\} = M$ *for any invertible linear map* P.

Proof. X and Y are linear maps from $\mathbb{R}^n \to \mathbb{R}^n$, so the problem of using matrix multiplication to generate the space of matrices is equivalent to using the monomials of X and Y (defined under composition) to span the space of all linear maps. Since linear maps are defined independent of a particular choice of basis, X and Y generates M regardless of the choice of

[1] In this paper, $A \subset B$ means $A \subsetneq B$.

basis. In other words, conjugation by a change of basis matrix P will not affect whether X and Y generate M. □

In the remainder of this paper, we use the notions that we have developed to analyze how efficiently a pair of real matrices X,Y can generate M, the vector space of real matrices. Section 2 presents the Monomial Spanning Algorithm and details the theoretical upper and lower bounds for $d(X, Y)$ over any field. Section 3 explores how we can place various conditions on the matrices X,Y in order to develop stronger statements about when and how quickly pairs of matrices with particular forms will span M. Section 4 introduces generic matrices and includes a proof that any pair of generic matrices generates $M = \mathbb{C}_{n\times n}$, which is the main result of our paper. Section 5 discusses the numerical tools we used to get an intuition about how X and Y generate M. Finally, Section 6 outlines how probabilistic methods were used to explore a variety of pairs of real matrices and we present a pair of matrices that generate M with $d(X, Y) < n$.

2. The Algorithmic Approach

This section develops an algorithm to compute the growth function of two matrices X and Y. Two algorithms will be presented. The first is a greedy algorithm that lacks a termination condition when $\text{Gen}\{X, Y\} \subset M$. The second algorithm adds the needed termination condition. Upon analysis, this termination condition gives lower and upper bounds for δ_n and Δ_n, respectively.

The first algorithm is based on the greedy approach. The greedy approach generates a basis for a vector space by maintaining a linearly independent set and adding any vector (or specifically in our case, a matrix) that preserves independence. When no such vectors exist, a basis has been found. In applying this approach to this problem, we need to impose an order on the vectors (the matrices) to ensure that the basis chosen is the most efficient. A natural choice for an ordering is the partial order based on the degrees of the monomials. However, for a complete description of the algorithm, we need a total order. By extending to a total order, a minimal basis of monomials with respect to the total order will still be minimal with respect to the the partial order. The natural total ordering is the lexicographic ordering on the monomials when treated as strings, as seen in the following definition.

Definition 2.1. The **lexicographic ordering** of the monomial terms of X and Y is the sequence

$$L = (I, X, Y, XX, XY, YX, YY, XXX, XXY, XYX, XYY, YXX, YXY, \ldots)$$

L is a linear ordering on the monomials of X and Y. Formally, it will impose the relation $<_L$ on the set of strings formed using the alphabet $\{X, Y\}$. I denotes the empty string and thus corresponds to the empty matrix product.

The following lemma captures the only essential property that we will be using of the lexicographic ordering aside from the fact that it generalizes the partial ordering based on degrees.

Lemma 2.2. *Monomial multiplication preserves order over L: if $u, v \in L$ with $u <_L v$, then $wu <_L wv$, $uz <_L vz$ and $wuz <_L wvz$ for all $w, z \in L$.*

Proof. By induction is enough to show the claim for $w, z \in \{X, Y\}$. That $Xu <_L Xv$ and $Yu <_L Yv$ follows from the definition of L. That $uX <_L vX$ and $uY <_L vY$ follows similarly. □

The following notation will be used. For $i \geq 0$, let l_i be the $(i+1)$-th element in the order L. Let $L_i = \{l_0, l_1, \ldots, l_i\}$. For a finite $S \subset L$ denote $|S|_L = \min\{i : S \subseteq L_i\}$ and let $\deg(S)$ be the degree of the maximum degree term in S. $\deg(S)$ and the more general $|S|_L$ will be the measures of efficiency. As not all X, Y generate M, we will focus on how efficiently sets of monomials can span L. For inductive purposes, we will also need to examine how efficiently sets of monomials can span the L_i. Formally, these measures will be $s_i = \min\{|S|_L : \mathrm{Span}(S) \supseteq L_i, S \subset L, |S| < \infty\}$ and d_i defined analogously with respect to the degrees of sets, $\deg(S)$. Define s_∞ and d_∞ analogously for $L_\infty = L$. As L resides in the n^2-dimensional M, it should be clear that $s_i, d_i < \infty$ as well as $s_\infty, d_\infty < \infty$.

With the notation in place, we can consider the first algorithm. It will greedily construct sets achieving the efficiencies s_i. The algorithm is the inductive definition of the following sets.

Definition 2.3. Let $S_0 = \{I\}$. For each $i > 0$, define

$$S_i = \begin{cases} S_{i-1} \cup \{l_i\} & l_i \notin \mathrm{Span}(S_{i-1}) \\ S_{i-1} & \text{else} \end{cases}$$

Some immediate properties of the S_i are that $S_i \subseteq S_j$ for $i < j$, $S_i \subseteq L_i$ and $\mathrm{Span}(S_i) = \mathrm{Span}(L_i)$. The following proposition establishes the key properties of the S_i.

Proposition 2.4. *For $i \geq 0$, $|S_i|_L = s_i$ and $\deg(S_i) = d_i$.*

Proof. The proof is by induction. We first consider the s_i.

Case $i = 0$: $S_0 = L_0$ so $|S_0|_L = 0 = s_i$.

Case $i > 0$: We condition on s_i. Suppose $s_i = i$. By the properties of S_i from above, it must be that $|S_i|_L = i = s_i$ by definition of s_i as a minimum.

Suppose $s_i = j < i$. Then there is some $S \subseteq L_j$ with $\mathrm{Span}(S) \supseteq L_i$. Then $\mathrm{Span}(L_j) = \mathrm{Span}(L_i)$ and so $S_i = S_j$ by the construction. By induction, $|S_j|_L = s_j$. By definition of the s_i, $s_j \leq s_i$. As $|S_i|_L \geq s_i$ and $S_i = S_j$ it must be that $|S_i|_L = s_i$.

The claim about the d_i follows as $\deg(S_i) > d_i$ implies $|S_i|_L > s_i$. □

This proposition shows that the S_i most efficiently span the L_i. As M is finite dimensional, there is some k for which $\mathrm{Span}(L_k) = \mathrm{Span}(L)$ and so S_k is an optimal set of monomials spanning L, implying $s_\infty = s_k$. Further, from the s_i, the d_i and the growth function can be calculated. While it is clear $k \leq n^2$, if $\mathrm{Gen}\{X, Y\} \subset M$ then $k < n^2$ and so there is no obvious termination condition to know when $s_k = s_\infty$ as no brute-force check of the infinitely many s_i is possible. We next provide the framework for providing the termination condition, starting with the following key observation.

Lemma 2.5. *Consider S_i and $l_j \notin \mathrm{Span}(S_i)$ with a decomposition into words $l_j = uwv$, where u, w and v are words over the alphabet $\{X, Y\}$ and w is non-empty. If $w \notin S_i$, then there is some $k < j$ so $l_k \notin \mathrm{Span}(S_i)$. That is, the minimum k with $l_k \notin \mathrm{Span}(S_i)$ is such that for any non-empty subword w of l_k, $w \in S_i$.*

Proof. To find the $k < j$ with $l_k \notin \text{Span}(S_i)$, we condition on various cases.

Case $w \notin \text{Span}(S_i)$: By Proposition 2.2 we have that $w <_L l_j$ in L. Therefore, $l_k = w \notin \text{Span}(S_i)$ for some $k < j$ and the claim is established.

Case $w \in \text{Span}(S_i)$, $w \notin S_i$: Take the minimum m such that $w \in \text{Span}(S_m)$. This is possible as $w \in \text{Span}(S_i)$ and $S_m \subseteq S_i$ for $m < i$. Thus the minimum such m has $m < i$. As w is a non-empty word over the alphabet $\{X, Y\}$, $m > 0$ as $l_0 = I$ and I is the empty word. As $w \notin \text{Span}(S_{m-1})$ then $S_m = S_{m-1} \cup \{l_m\}$. As $w \notin S_i \supseteq S_m$ then $w \neq l_m$ and so $w \neq_L l_m$. If $w <_L l_m$ then $w \in L_p \subseteq \text{Span}(S_p)$ for some $p < m$, which contradicts that m was chosen as a minimum. Thus $w >_L l_m$.

Thus, as $w \in \text{Span}(S_m)$ and $w >_L l_m$, $w = \sum_{l_s \in S_m} a_s \cdot l_s$ where $a_s \in \mathbb{R}$ and $l_s <_L w$. Applying matrix multiplication, $l_j = uwv = \sum_{l_s \in S_m} a_s \cdot ul_sv$. As $l_j \notin \text{Span}(S_i)$, there must be some $l_r \in S_m$ such that $ul_rv \notin \text{Span}(S_i)$. Let $l_k = ul_rv$. As $l_r <_L w$ then by Proposition 2.2, $l_k = ul_rv <_L uwv = l_j$ and so the claim is established. □

By framing the previous lemma in its contrapositive form and adding some strength, we get the next corollary.

Corollary 2.6. *For S_i such that $\text{Span}(S) \not\supseteq L$ then the minimum j such that $l_j \notin \text{Span}(S_i)$, it must be that l_j lies within the set of monomials $\{Xw, Yw : w \in S_i\}$ and thus $\deg(l_j) \leq \deg(S_i) + 1$.*

Proof. Take the minimum j so that $l_j \notin \text{Span}(S_i)$. By Proposition 2.5, all non-empty subwords of l_j must be in S_i. As the empty word $I \in S_0 \subseteq S_i$, it is in fact that all subwords of l_j are in S_i. Thus, if $l_j = Xw$ for some $w \in L$ then $w \in S_i$ and similarly if $l_j = Yw$. As $\deg(l_j) = \deg(w) + 1$ and $w \in S_i$, $\deg(l_j) \leq \deg(S_i) + 1$. □

The sequence of S_i are not strictly increasing sets and this corollary describes how to bypass the non-increasing aspects of the sequence. Specifically, for S_i not spanning L the next j such that $S_i \subset S_j$ is $j = \min_j\{l_j : l_j \notin \text{Span}(S_i), l_j = Xw \text{ or } l_j = Yw, w \in S_i\}$. As $\{Xw, Yw : w \in S_i\}$ is a finite set, the minimum such j can be computed effectively. As we are only interested in cases for which $S_{i+1} \supset S_i$ we can then use the computation for j from above to get the second algorithm which defines the largest strictly increasing subsequence of the S_i.

Algorithm 1 Monomial Spanning Algorithm

$T_0 = \{I\}$
$i \leftarrow 0$
while $\exists A \in \{Xw, Yw : w \in T_i\} : A \notin \text{Span}(T_i)$ **do**
  $j \leftarrow \min_j\{l_j : l_j \notin \text{Span}(T_i), l_j = Xw \text{ or } l_j = Yw, w \in T_i\}$
  $T_{i+1} \leftarrow T_i \cup \{l_j\}$
  $i \leftarrow i + 1$
end while
$T_\infty \leftarrow T_i$

There are some fairly easy facts to deduce from this algorithm. Most importantly, the algorithm terminates and yields an optimal set of monomials spanning L. This is because each iteration of the while-loop of the algorithm adds an independent matrix to T_i. As M

is of dimension n^2 this can occur at most $n^2 - 1$ times (as we start with T_0) and thus the algorithm terminates with T_∞ spanning L. Further, from Corollary 2.6, it follows that the T_i are the largest strictly increasing subsequence of the S_i and thus are still the optimal sets for spanning the L_i and correctly compute the growth function. The next two important properties are given as propositions.

Proposition 2.7. $\deg(T_\infty) \leq n^2 - 1$

Proof. $\deg(T_0) = 0$ and by Corollary 2.6, $\deg(T_{i+1}) = \deg(T_i \cup \{l_j\}) \leq \deg(T_i) + 1$. As there are at most $n^2 - 1$ iterations of the while loop, $\deg(T_\infty) \leq n^2 - 1 + \deg(T_0) = n^2 - 1$. □

Proposition 2.8. *For $n \times n$ matrices, the Monomial Spanning Algorithm runs in an amount of time polynomial in n.*

Proof. There are $O(n^2)$ iterations of the while-loop. As $|T_i| \leq n^2$, calculating $\min_j\{l_j : l_j \notin \mathrm{Span}(T_i), l_j = Xw \text{ or } l_j = Yw, w \in T_i\}$ takes $O(n^2 f(n))$ steps, where $f(n)$ is the amount of time it takes to test for linear independence of a size $O(n^2)$ set of $n \times n$ matrices. As the usual algorithm for testing independence in an explicit vector space takes polynomial time, this yields the claim. □

These properties then give an algorithmic proof of an upper bound and a lower bound for the degrees of the monomials when X, Y span M.

Theorem 2.9. *If X, Y generate M then they generate it with monomials of degree at most $n^2 - 1$. That is, $\Delta_n \leq n^2 - 1$.*

Proof. If X and Y generate M then $\mathrm{Span}(T_\infty) = \mathrm{Span}(L) = M$. The theorem follows from Proposition 2.7. □

Theorem 2.10. *If X and Y generate M then they generate it with monomials of degree at least $\lg(n^2 + 1) - 1$.*[2] *That is, $\delta_n \geq \lg(n^2 + 1) - 1$.*

Proof. If X and Y generate M then $\mathrm{Span}(T_\infty) = \mathrm{Span}(L) = M$. As M is n^2 dimensional it must be that $|T_\infty| = n^2$. For a size n^2 subset S of L, the minimum $|S|_L$ possible is $n^2 - 1$ as achieved by L_{n^2-1}. The set of monomials of degree at most d is of size $\sum_{i=0}^{d} 2^i = 2^{d+1} - 1$. Therefore the degree of T_∞ is at least the least d so that $2^{d+1} - 1 \geq n^2$. In other words, $\deg(T_\infty) \geq \log(n^2 + 1) - 1$. As T_∞ is the optimal set for generating M, the theorem follows. □

These theorems put upper and lower bounds on $d(X, Y)$ for any X, Y which have $\mathrm{Gen}\{X, Y\} = M$. It is important to notice that no properties specific to $\mathbb{R}$, the form of X, Y, or even that X, Y are matrices, were used. However, this generality seems to come at the cost of tightness as we know of no examples that come close to these bounds. In the next section we explore various conditions on the form of X, Y that allow us to state tighter results about their degree when they span M.

3. Special Cases

This section sacrifices some generality to obtain tighter results on the efficiency of two matrices spanning M. The first result is as follows.

[2] lg denotes logarithm base 2.

Proposition 3.1. *Let D and A be $n \times n$ matrices where D is diagonal with distinct non-zero diagonal entries, and A has all non-zero entries. Then A, D generate M with monomials of degree at most $2n-2$. That is, $d(A, D) \leq 2n-2$.*

Proof. For this proof we denote for a matrix N that N_{ij} is the entry in N in the i-th row and j-th column.

The first observation of this proof is that because D has distinct non-zero diagonal entries, they can be zeroed out individually. Specifically, the matrices $D - D_{ii}I$ are diagonal with exactly one zero diagonal entry at the i-th row and column. Thus, this allows row and column operations that can zero-out single rows and columns at a time. As D is diagonal, if we zero-out all but one row (or column) we are left with a multiple of a matrix unit E_{ii}. Thus, we obtain the following formula $E_{ii} = \frac{1}{\prod_{k \neq i}(D_{ii} - D_{kk})} \prod_{k \neq i} (D - D_{kk}I)$ and so E_{ii} is generated with degree $n-1$ monomials as the fraction is just a scalar and there is a product of $n-1$ linear terms.

For A, D to generate M it is necessary and sufficient for A, D to generate the matrix units E_{ij}. As we already have the matrix units along the diagonal, it suffices to obtain the rest.

Claim. *A, D generates the matrix unit E_{ij}, $i \neq j$, with monomials of degree at most $2n-2$.*

Proof. As we already have the E_{ii} we can notice that $E_{ii}AE_{jj}$ is just $A_{ij}E_{ij}$. By normalizing, we can then generate the matrix units E_{ij}. However, this requires degree $(n-1)+1+(n-1) = 2n-1$ monomials.

To achieve degree $2n-2$, we must do something slightly more clever. We achieve this by zeroing out all but one row but leave two of the columns untouched. This yields a matrix with two non-zero entries and is thus a linear combination of two matrix units. As we already have the diagonal matrix units, we can choose one of these matrix units to be diagonal and subtract it off to yield the desired matrix unit after normalization. Formally,

$$E_{ij} = \frac{1}{A_{ij} \prod_{k \neq i,j}(D_{jj} - D_{kk})} \left(E_{ii} A \prod_{k \neq i,j} (D - D_{kk}I) - \left(A_{ii} \prod_{k \neq i,j} (D_{ii} - D_{kk}) \right) E_{ii} \right)$$

The above equation has terms of maximum degree $(n-1) + 1 + (n-2) = 2n-2$ as E_{ii} is generated with degree $n-1$. □

As A and D generate all matrix units using monomials of degree at most $2n-2$ they generate M with such monomials. □

We do not have a formal proof that this bound is tight, but the numerical results of Section 5.2 suggest that it is.

Notice that the requirement that A has all non-zero entries is crucial. For example, suppose A was itself diagonal. Then all monomials would also be diagonal. Therefore, A, D could never generate M. However, we can still say something interesting if A has *enough* non-zero entries. Before making this precise, we consider some other special cases that will lead into the generalization of the above proposition.

Proposition 3.2. *Let P, E_{kl} be $n \times n$ matrices where P is a transitive permutation matrix and E_{kl} is some matrix unit. Then P, E_{kl} generate M with monomials of degree at most $2n-1$. That is, $d(P, E_{kl}) \leq 2n-1$.*

Proof. Consider the monomials $P^i E_{kl} P^j$ for $0 \leq i, j \leq n-1$. As P is a transitive permutation matrix, all matrix units appear in this list. As the maximum degree here is $2n-1$ and all matrix units appear, we have that P, E_{kl} generate M with this degree. □

While we did not explore the above case experimentally, we did consider when we specifically have a diagonal matrix unit. For this case we have the following tighter result.

Corollary 3.3. *Let P, E_{kk} be $n \times n$ matrices where P is a transitive permutation matrix and E_{kk} is a diagonal matrix unit. Then P, E_{kk} generate M with monomials of degree at most $2n-2$. That is, $d(P, E_{kk}) \leq 2n-2$.*

Proof. Again, consider the monomials $P^i E_{kk} P^j$ but now for $0 \leq i + j \leq 2n-3$. The only matrix unit not appearing in this list is $E_{ll} = P^{n-1} E_{kk} P^{n-1}$, for some l. However, as we have all of the other diagonal matrix units and the identity matrix I, $E_{ll} = I - \sum_{i \neq l} E_{ii} = I - \sum_{0 \leq i \leq n-2} P^i E_{kk} P^i$. Thus, by generating all matrix units with degree at most $2n-2$, we have that P, E_{kk} generate M with this degree. □

Again, we have no proof that the above bound is tight, but the numerical results of Section 5.1 suggest that it is.

To achieve a generalization of Proposition 3.1, we want to deal with more than just matrix units and to instead have diagonal matrices with distinct non-zero diagonal entries alongside the transitive permutation matrices. We can get back into that case by considering the following lemma.

Lemma 3.4. *Let D be an $n \times n$ diagonal matrix with distinct non-zero diagonal entries. Then D generates the space of diagonal matrices with monomials of degree $n-1$.*

Proof. As the space if diagonal matrices is dimension n, it suffices to exhibit n linearly independent diagonal matrices. The set $\{I, D, \ldots, D^{n-1}\}$ is linearly independent as D has distinct non-zero diagonal entries and thus the Vandermonde determinant of the n-dimensional diagonal vectors of these matrices is non-zero. □

We can then consider what happens when we add a transitive permutation matrix.

Proposition 3.5. *Let P, D be $n \times n$ matrices where P is a transitive permutation matrix and D is a diagonal matrix with distinct non-zero diagonal entries. Then P, D generate M with monomials of degree at most $2n-2$. That is, $d(P, D) \leq 2n-2$.*

Proof. Consider the monomials $\{P^i D^j : 0 \leq i, j \leq n-1\}$. As P is transitive, it moves the non-zero diagonal of D around in a non-overlapping way. Thus, $P^i D^j$ and $P^i D^k$ have no entries where they are both non-zero for $j \neq k$. Thus, a linear dependency on the set of monomials would imply some linear dependency within some set $\{P^i D^j : 0 \leq j \leq n-1\}$ for fixed i. But this cannot occur by the previous lemma as P is invertible. As this set is of n^2 monomials and M is n^2 dimensional, they span M (and the maximum degree is $2n-2$). □

While the generation of the diagonal matrices is a tight bound as we only have one matrix, the above construction combining P and D does not seem to give a tight result which is unlike the earlier constructions in this section. See Section 5.3 for numerical results that suggest this result is not tight.

Now we can consider a looser version of Proposition 3.1. Define the *support* of a matrix as the set of coordinates where it has non-zero entries. We will say that a matrix A has

transitive support if its support is the superset of the support of some transitive permutation matrix. Thus, if D is a diagonal matrix with distinct non-zero entries and A is a matrix with transitive support, the proof of Proposition 3.1 shows that there is some transitive permutation matrix $P \in \text{Gen}\{A, D\}$. P, D generate M by Proposition 3.5, and thus so do A, D. Unfortunately, the degrees in this composed construction are worse than those guaranteed by Theorem 2.9, but knowing that $\text{Gen}\{A, D\} = M$ then allows the following result as a corollary.

Corollary 3.6. *Let D, A be $n \times n$ matrices where D is diagonal with distinct non-zero diagonal entries, and A has transitive support. Then A, D generate M with degree at most $n^2 - 1$. That is, $d(A, D) \leq n^2 - 1$.*

This result, as a generalization of Proposition 3.5, does not seem to give a tight bound on the degree.

The previous section used an algorithmic approach that ignored the underlying field of the matrices, that the vectors actually were matrices and used no specific forms to the matrices. This section explicitly used special matrix forms to establish tighter results than the general algorithm. However, it is interesting to note it is still the case that at no point was the underlying field $\mathbb{R}$ used. Thus, the results of this section hold for any field. In the next section, the underlying field (changed to $\mathbb{C}$) will be explicitly used to show that these special cases imply that "most" matrices span M.

4. Genericity and the Main Result

Specifying a complete list of necessary and sufficient conditions for X, Y to generate still remains, for the purposes of this project, an open problem. However, the power of the examples illustrated thus far actually provides a great deal of information. Instead of considering the entire space M from which we choose X and Y, we will restrict our choice to some set $M - S$ where $S \subset M$ has measure zero. We will refer to such restricted matrices as *generic matrices* and we will see that under certain specifications of S we can generate M using any generic pair of matrices.

Definition 4.1. A set $S \subset \mathbb{R}^n$ has **measure zero** if, for every $\epsilon > 0$, S can be covered by countably many n-rectangles with total volume less than ϵ ([3] p.91).

It is important to note that under this definition countable sets always have measure zero, as do manifolds of dimension less than n, thus the ambient space is important for measure theoretic purposes. For example, a line does not have measure zero in $\mathbb{R}^1$, but it does in $\mathbb{R}^2$. Thus any submanifold (or countable union thereof) of dimension $d \leq n - 1$ will have measure zero when viewed as sitting inside of $\mathbb{R}^n$.

Definition 4.2. Let M be the space of $n \times n$ matrices over an arbitrary field. A **generic matrix** is any matrix $X \in M - S$ where $S \subset M$ (closed) has measure zero. Accordingly, since the complement of a closed measure zero set is an open dense set ([2] p.118), generic matrices are dense in M once S is specified.

S is often taken to be the solution set of some set of equations defined on M. We can imagine using S to exclude some hypersurface of "bad" matrices, which leaves behind a dense set of "good" matrices to choose from. Before we proceed to outline our main result, let us illustrate our notion of a generic matrix with two concrete examples.

Lemma 4.3. *The set of matrices with distinct eigenvalues is an open dense subset of* $M = \mathbb{C}_{n\times n}$.

Proof. If $A \in \mathbb{C}_{n\times n}$ then there exists a $Q \in GL_n(\mathbb{C})$ such that $T = Q^{-1}AQ$ is upper triangular. Suppose the diagonal entires of T are d_{ii}. Choose numbers $k_1, \ldots, k_n$ such that $|k_i| < \epsilon$ for all i and so that each of the $d_{ii} + k_i$ are distinct. If K is the diagonal matrix with diagonal entries k_i then by construction $T + K$ has distinct eigenvalues. Thus $Q(T + K)Q^{-1} = A' = QTQ^{-1} + QKQ^{-1}$ has distinct eigenvalues. We also note that $||A' - A|| = ||QKQ^{-1}|| = \max_{i,j} |QKQ^{-1}|_{ij}$ and thus by choosing $\epsilon > 0$ small enough we can approximate A with a sequence of matrices with distinct eigenvalues (i.e. the set is dense).

To show that the set is open, we show that the complement is closed [1]. Suppose $A_k \to A$ is a sequence of matrices with at most $n-1$ distinct eigenvalues. If $p_k(\lambda) = \det(\lambda I - A_k)$ is the characteristic polynomials of A_k, then there is some λ_k such that $p_k(\lambda_k) = p_k'(\lambda_k) = 0$. Suppose $p_k(\lambda) = \det(\lambda I - A)$ is the polynomial of the limit matrix. Since the determinant is a continuous function, $p_k \to p$. Since the coefficients of p_k are determined by the entries of A_k, which converges to A, the coefficients are uniformly bounded. Consequently, $|p_k(\lambda) - \lambda| \leq M \sum_{j=0}^{n-1} |\lambda|^j$ and since $p_k(\lambda_k) = 0$, $|\lambda_k|^n \leq M \sum_{j=0}^{n-1} |\lambda_k|^j$ so $\{\lambda_k : k \in \mathbb{N}\}$ is bounded. By the Bolzano-Weierstrass theorem, we can find a convergent subsequence such that $\lim_{j\to\infty} \lambda_{k_j} = \mu$. Since $\lim_{j\to\infty} p_{k_j}(\lambda_{k_j}) = \lim_{j\to\infty} p'_{k_j}(\lambda_{k_j}) = 0$, we have that $p(\mu) = p'(\mu) = 0$ and thus μ is an eigenvalue with algebraic multiplicity at least two, i.e. A has at most $n-1$ distinct eigenvalues. This proves that the complement is closed and thus the set is open. □

It is important to note that Lemma 4.3 is not true if $\mathbb{C}_{n\times n}$ is replaced by $\mathbb{R}_{n\times n}$, since real matrices are not always conjugate to ones in triangular form. For this reason we consider generic complex matrices. The next lemma is true over both fields.

Lemma 4.4. *The set of matrices with all non-zero entries is an open dense subset of* M *over* $\mathbb{R}$ *or* $\mathbb{C}$.

Proof. Let $X \in M$. Suppose $X_{ij} = 0$ is some zero entry, then $X' = X + \epsilon E_{ij}$ can be made arbitrarily close to X for $\epsilon > 0$. Repeating for finitely many entries, any X can be approximated by a sequence of matrices with all non-zero entries and thus the set is dense. To prove that this is an open dense set, we consider the projection map $\pi_{ij} : M \to \mathbb{R}$ (or $\mathbb{C}$) defined by $\pi_{ij}(X) = X_{ij}$. π_{ij} is clearly continuous, thus $\pi_{ij}^{-1}(0)$ is a closed set. Since the finite union of closed sets is closed, running over $1 \leq i, j \leq n$, the set of matrices with at least one zero entry is closed, and thus its complement is open. □

We are now able to prove our main result.

Theorem 4.5. *Two generic matrices* X, Y *generate* $M = \mathbb{C}_{n\times n}$.

Proof. A pair of matrices $(X, Y) \in M \times M$ are generic if they are defined on some open dense subset in the product topology. By the above two lemmas the set S_1 of matrices with non-distinct eigenvalues and the set S_2 of matrices with zero-entries have measure zero. Since conjugation by a linear map sends measure zero sets to measure zero sets we have that conjugates of generic sets are generic. If $Q = M - S_1$ and $N = M - S_2$, then for $X \in Q$ there exists an invertible P such that $P^{-1}XP = D \in P^{-1}QP$ is diagonal. Choosing $Y' \in P^{-1}NP \cap N$ a non-zero matrix after conjugation, we can then apply Corollary 3.6 to attain a generating pair of matrices. Using Proposition 1.8 from the introduction and

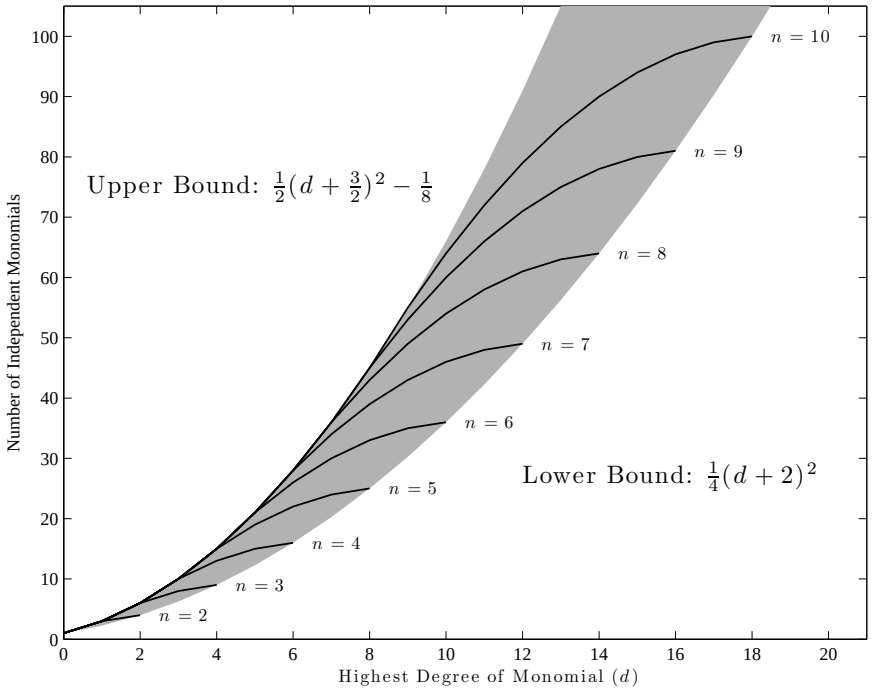

FIGURE 1. Transitive Permutation Matrix and E_{11}

conjugating by P^{-1} we have that $X = PDP^{-1} \in Q$ and $Y = PY'P^{-1} \in N \cap PNP^{-1}$ generate M as well. □

5. Numerical Investigations

The previous section gave us a theoretical condition for when two complex matrices X and Y generate $M = \mathbb{C}_{n\times n}$. In this section, we attack the problem of generating real matrices with computational tools in an attempt to infer patterns and form conjectures. MATLAB is used to implement the algorithm and produce corresponding plots. The files and their usage is described in the appendix.

The algorithm essentially is an implementation of the Monomial Spanning Algorithm described in Section 2. We use a recursive procedure that extends an assumed independent set to a larger one. The base case is the set of degree zero monomials, which is just I. To find linearly independent sets of higher degree we extend by considering X and Y times the independent set from the previous degree. We now compute some special cases considered in Section 3.

5.1. **Transitive Permutation and One Non-Zero Entry.** Perhaps the clearest way to generate $M = \mathbb{R}_{n\times n}$ is to use the following two matrices:

$$X = \left[\begin{array}{c|c} 0_{1\times n-1} & 1 \\ \hline I_{n-1\times n-1} & 0_{n-1\times 1} \end{array}\right], \qquad Y = \left[\begin{array}{c|c} 1 & 0_{1\times n-1} \\ \hline 0_{n-1\times 1} & 0_{n-1\times n-1} \end{array}\right] = E_{11}.$$

Here X is a transitive permutation matrix and obeys $X^n = I$. By letting X operate on the right and left of Y we can transitively swap columns and rows respectively thereby transforming E_{11} to E_{ij} for any $1 \le i, j \le n$ we like. Thus the highest needed degree

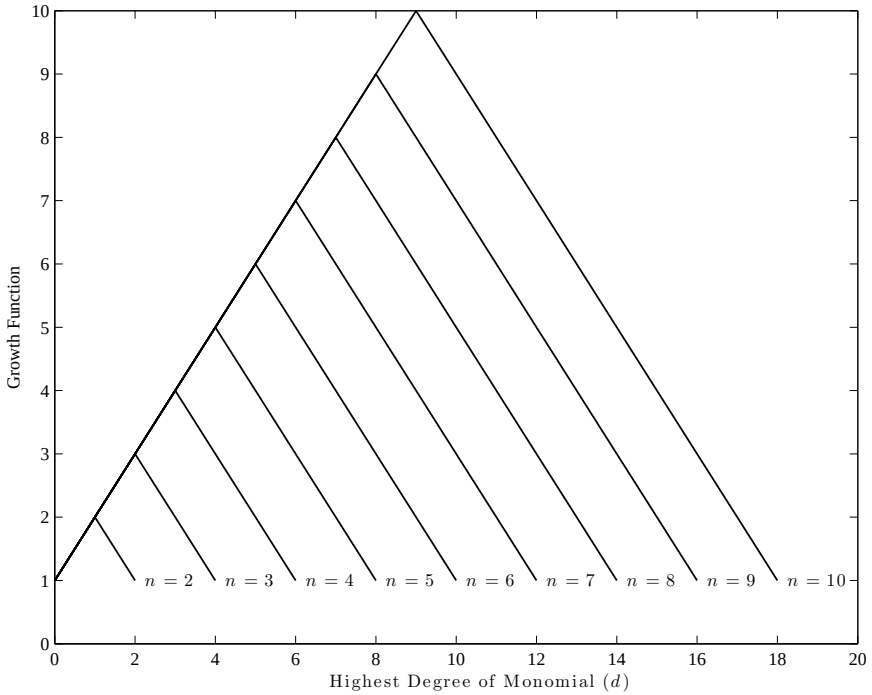

FIGURE 2. Transitive Permutation Matrix and E_{11}: Growth Function

monomial we possibly might need to consider is $X^{n-1}YX^{n-1}$ or $d = 2n-1$. Figure 1 depicts the number of independent monomials less than or equal to a given degree. We see that spanning is achieved for $d = 2n-2$, which agrees with our estimated upper bound. Since n^2 is the number of elements required for spanning we use $d = 2n-2 \Rightarrow n = d/2+1 \Rightarrow n^2 = 1/4(d+2)^2$ to bound the terminal points for each curve. We also note that up to certain values of d, the number of independent monomials of a given degree agree for a range of n's. Using a quadratic regression we find the equation of the projected matching to be $1/2(d+3/2)^2 - 1/8$. Although the lower bound comes from a theoretical trend of spanning degree $d = 2n-2$, the observed and projected matching has no well understood origin.

Figure 2 shows the growth function for this pair of X and Y. The growth function provides the number of independent matrices for a given degree d of monomials. It can be interpreted as a discretized derivative of the number of independent monomials depicted in Figure 1. The regularity of the observed behavior suggests that the most rapid increase in the dimension of the spanned subspace occurs around $d = n-1$ or exactly half the degree required to achieve spanning. In particular this maximum growth happens to coincide exactly with the dimension of the matrix n. Thus for our example, as a function of degree, the maximum growth for a set of monomials occurs at $d = n-1$ with maximum value n.

5.2. **All Non-Zeros and Distinct Eigenvalues.** In Section 3 it was shown that if X is a matrix with all non-zero entries (for example, the ones matrix) and Y is a matrix with distinct diagonal entries (assume $1, \ldots, n$) then X and Y generate M with monomials of degree at most $d = 2n-2$. It is interesting to note that this is also the noted spanning degree for the previous case with a transitive permutation matrix and E_{11}. It is an interesting fact that both the number of independent monomials and the growth function are *identical* for these

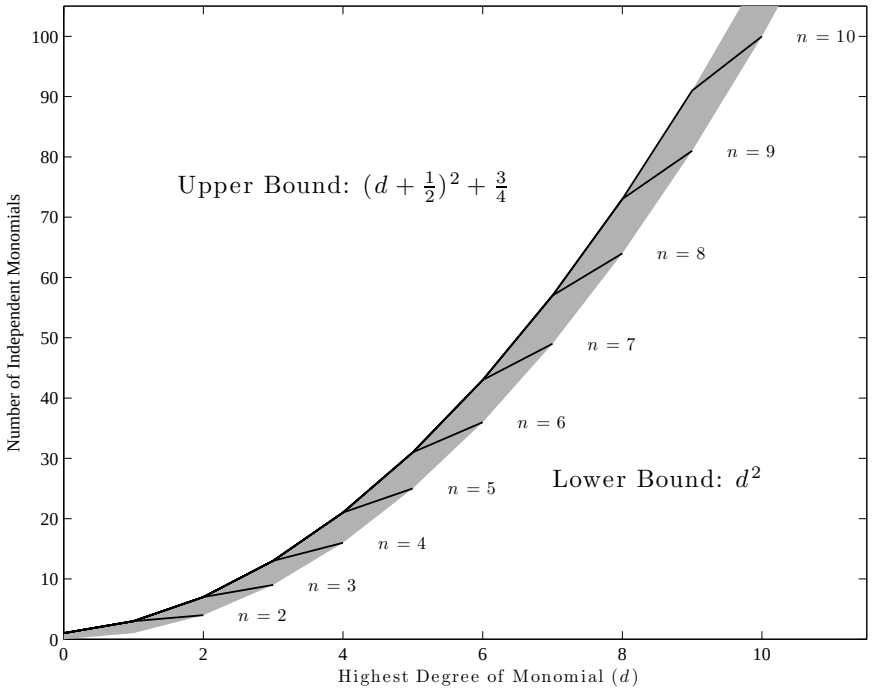

FIGURE 3. Transitive Permutation Matrix and Distinct Diagonal

two cases and as such we did not plot them. This match holds for all matrices up until $n = 8$ where machine tolerance issues cause round off error and we thus lose independence. The values of n that actually permit spanning are displayed in Table 1. Machine tolerance is understandably an issue because we anticipate that $d = 2 \times 8 - 2 = 14$ will be the minimal spanning degree for $n = 8$, but by construction the last most entry on the diagonal of Y is 8 and considering potentially the 14^{th} power implies that numbers as large as $8^{14} = 2^{42}$ may be encountered.

TABLE 1. Minimal Degree Needed to Span

$\boldsymbol{n}$	2	3	4	5	6	7
$\boldsymbol{d}$	2	4	6	8	10	12

5.3. **Transitive Permutation and Distinct Eigenvalues.** Although the previous two examples illustrated two very different pairs of matrices that exhibit the same growth function, here we consider a blend of these two sorts of matrices to achieve a different (but surprisingly related) growth function. Here we take X to be the same transitive permutation matrix we considered before, and let Y be a diagonal matrix with $1, \ldots, n$ running along the diagonal. Figure 3 illustrates the number of independent monomials less than or equal to a given degree. For these matrices $d = n$ achieves spanning for $n \leq 10$, with machine tolerance issues causing failure to span for larger n.

Now let us turn our attention to the growth function, limited to the accurate data runs for $n \leq 10$. The growth function is clearly different than the previous case, as this pair of matrices spans M more efficiently (i.e. achieves spanning with lower degree monomials).

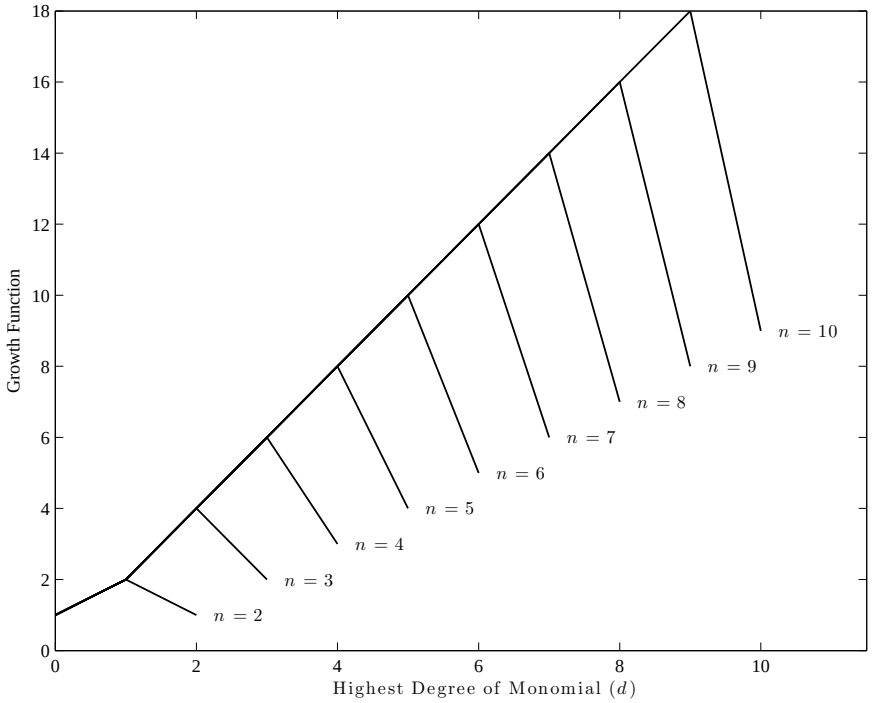

FIGURE 4. Transitive Permutation Matrix and Distinct Diagonal: Growth Function

However, notice that the maximum value occurs at $d = n - 1$ just as before. Furthermore, note that the maximum value is $2n - 2$, which is the degree required to span in the previous cases. Although the matrices were well-chosen for all three of these examples it is surprising that the growth functions are *dual* with respect to the minimal spanning degree and the maximum jump in dimension for degree $d = n - 1$. Whether or not this is just a remarkable coincidence is an open question worthy of future exploration.

6. Probabilistic Investigations

Since none of the examples investigated fall near the upper bound from Theorem 2.9 or the lower bound from Theorem 2.10, we used probabilistic sampling to explore a wider variety of matrices. In particular, we wrote a MATLAB script that calculates how many degrees of the monomials were needed to generate M from random matrices X and Y of certain forms.

For the cases we have analyzed, X and Y generate M most efficiently when X is a random matrix with integral entries and Y is a Jordan block. One interesting example of this form that we explored is where Y is a Jordan block with eigenvalue 0 and X is a matrix with ones along the two diagonals above the main diagonal and along the two diagonals below the main diagonal. X and Y of this form, for the case where $n = 5$, are shown in Figure 5.

The minimal degree d needed for the pair X,Y to generate M is shown in Table 2 for $2 \leq n \leq 19$. For $3 \leq n \leq 19$, the data fits the step function $d = \lceil n/2 \rceil + 2$ for $d > 2$. This is the first example we found where M is generated by the set of monomials of degree d for $d < n$.

$$X = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

FIGURE 5. X,Y where $n = 5$

TABLE 2. Minimal Degree Needed to Span

$\boldsymbol{n}$	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$\boldsymbol{d}$	2	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12

It is interesting to note that our theoretical bounds of $\mathrm{O}(n^2)$ and $\Omega(\log n)$ for the spanning degree may not be tight. All the concrete examples considered thus far span with degree $d = \Theta(n)$, but we have been unable to prove tighter bounds.

7. Division of Labor

- Justin Curry – Genericity and The Main Result, Numerical Investigations
- Michael Forbes – The Algorithmic Approach, Special Cases
- Matthew Gordon – Introduction, Probabilistic Investigations

Appendix A. Code

- `monomials.m` is the primary function that we use. It takes as input two square matrices X, Y of the same size and a degree d that is the highest degree of highest monomial considered. The output consists of two cell arrays, one consisting of the independent matrix monomials of degree $\leq d$, the other consisting of the "words" or symbolic representatives of the independent matrix monomials. The usage is `[mon words]=monomials(X,Y,d)`.
- `extendbasis.m` is the fundamental function that `monomials.m` relies on. It takes as input an assumed linearly independent set of matrix monomials and their corresponding words, along with a new set of matrix monomials and words. The output is a maximally linearly independent set that includes as many matrices from the new set while retaining independence.
- `testing.m` is the script used to collect data from `monomials.m`
- `plotmonomials.m` and `growthfun.m` are the base functions used to plot data. The former gives the number of independent monomials less than or equal to a given degree and the latter calculates the growth function as described in the introduction.

Since these latter two scripts are just used for executing a large number of examples and plotting the results, we only include `monomials.m` and `extendbasis.m` below.

A.1. **extendbasis.m.**

```
function [mon words] = extendbasis(mprev,wprev,mtest,wtest)
% ASSUMES THAT mprev AND wprev ARE ALREADY LINEARLY INDEPENDENT
% INPUT:  mprev   list of lower d monomials
```

```
%         wprev   list of words for lower d monomials
%         mtest   list of higher d monomials to test for linear dependence
%         wtest   list of words for said higher d monomials
%
% OUTPUT: mon     list of all linearly independent monomials of degree <= d
%         words   list of words for all monomials of degree <= d
%
% USAGE:  [mon words] = extendbasis(mprev,wprev,mtest,wtest)


% initialize
Dp = length(mprev); Dt = length(mtest);  % lengths of monomial lists
n = size(mprev{1},1);   % size of generating matrices
mon = cell(Dp+Dt,1);    % output list of monomials
words = cell(Dp+Dt,1);  % output list of words
Aprev = zeros(n^2,Dp);  % matrix whose columns are reshaped monomials

% copy lower d lists to output
for k=1:Dp
    mon{k} = mprev{k};
    words{k} = wprev{k};
    Aprev(:,k) = mprev{k}(:);  % setup matrix for testing linear dependence
end

i=0;
for k=1:Dt
    Atest = [Aprev mtest{k}(:)];  % append one monomial at a time
    % save monomial to output list if linearly indepenedent from others
    if (rank(Atest)==size(Atest,2)),  % check rank
        Aprev = Atest;
        i=i+1;
        mon{Dp+i} = mtest{k};
        words{Dp+i} = wtest{k};
    end
end

% delete empty entries from output lists
mon(cellfun(@numel,mon)<1)=[];
words(cellfun(@numel,words)<1)=[];
```

A.2. **monomials.m.**

```
function [mon words] = monomials(X,Y,d)
% INPUT:  X,Y     generating matrices (must be square, same size)
%         d       integer, highest degree for which to generate monomials
%
% OUTPUT: mon     cell array of all monomials of degree <= d
```

```
%         words   cell array of words for all monomials of degree <= d
%
% USAGE:  [mon words] = monomials(X,Y,d)

n = size(X,1);  % dimension of generating matrices

if (d==0)  % base case (identity)

    mon{1} = diag(ones(1,n));
    words{1} = 'I';

else  % recursively append linearly independent monomials to list

    [mprev wprev] = monomials(X,Y,d-1); % recursive call to lower d list
    dprev = length(mprev);
    mnext = cell(2*dprev,1);
    wnext = cell(2*dprev,1);

    % generate the higher d list
    for k=1:dprev
        mnext{k} = X*mprev{k};
        mnext{k+dprev} = Y*mprev{k};
        if (wprev{k} == 'I'),
            wnext{k} = 'X';
            wnext{k+dprev} = 'Y';
        else
            wnext{k} = ['X' wprev{k}];
            wnext{k+dprev} = ['Y' wprev{k}];
        end
    end

    % append linearly independent higher d monomials to lower d list
    [mon words] = extendbasis(mprev,wprev,mnext,wnext);

end


end
```

References

1. Simon Brendle, *Mathematics 53H*, [Online; accessed 19-April-2008; only used for proof that the set of matrices with distinct eigenvalues is open.].
2. Anthony W. Knapp, *Basic real analysis*, Birkhäuser, 2005.
3. James R. Munkres, *Analysis on manifolds*, Westview Press, 1991.