Reliable method for the evaluation of commuting matrices

P. Carrette

Department of Electrical Engineering
Linkping University, S-581 83 Linkping, Sweden
WWW: http://www.control.isy.liu.se
Email: carrette@isy.liu.se

March 5, 1998

Report no.: LiTH-ISY-R-2012

Technical reports from the Automatic Control group in Linkping are available by anonymous ftp at the address ftp.control.isy.liu.se. This report is contained in the compressed postscript file 2012.ps.Z.
Reliable method for the evaluation of commuting matrices

P. Carrette - carrette@isy.liu.se

Department of Electrical Engineering, Linköping University
S-58183 Linköping, Sweden

Abstract

In this paper, we study the characteristics of matrices that commute with a given matrix $A$. It appears that these are related to the null space of a well-structured matrix. More precisely, any matrix $R$ that commutes with $A$ has its vectorized column in the null space of that well-structured matrix. Hence, the problem of finding such matrix $R$ is equivalent to that of obtaining a representation of a basis for that space. From a numerical point of view, we propose a low cost and robust procedure for computing a basis for the “commuting” null space.

Keywords: matrix commutation, null space, Kronecker product

1 Introduction

In this communication, we consider the problem of characterizing the set of matrices $R$ that commute with a given $n$-dimensional complex matrix $A \in \mathbb{C}^{n \times n}$. Furthermore, we are interested in doing this in a numerically reliable way whatever the matrix $A$ is, i.e. possibly nondiagonalizable.

The condition for the matrix $R$ to commute with $A$ is simply written as

$$RA = AR$$

In case the matrix $A$ is diagonalizable, an easy expression is found for $R$. Indeed, first this matrix is identical to $V \Lambda V^{-1}$ where $\Lambda$ is a diagonal matrix containing the eigenvalues and $V$ has the corresponding eigenvectors as columns. Then, it is not difficult to see that any matrix $R$ of the form

$$R = V \text{diag}(r) V^{-1} \quad \text{with} \ r \in \mathbb{C}^n$$

satisfies the commuting condition. In fact, this expression completely characterizes the set of matrices $R$ when the given $A$ has distinct eigenvalues.
For what concerns the numerics, the result is based on the eigenvalue decomposition of the matrix $A$. Although the matrix eigensubspaces are robustly evaluated, this characterization becomes numerically disastrous in case the eigenvector matrix $V$ is close to singular. The reason for this is simply that the matrix $R$ is expressed in terms of the inverse of this eigenvector matrix. The situations where this latter matrix becomes singular exactly coincide with those for which the matrix $A$ is anymore diagonalizable, i.e., presence of Jordan blocks preventing us from writing $A = VAV^{-1}$.

This note is devoted to make use of the more general characterization of commuting matrices. It is derived with the help of the “vec” operator [2, chap. 4] as:

$$(A^T \otimes I_n - I_n \otimes A) \, \text{vec}(R) = 0$$

where $\otimes$ denotes the Kronecker product and $I_n$ stands for the identity matrix in $\mathbb{C}^n$. This equation means that, for having $R$ commuting with the given $A$, the column vec($R$) has to lie in the null space $Q$ of the matrix $(A^T \otimes I_n - I_n \otimes A)$. Then, by denoting $Q \in \mathbb{C}^{n^2 \times q}$ (with $q = \dim(Q) \geq n$) a matrix whose columns constitute a basis for that null space, any matrix $R$ has the following form

$$R = \text{vec}^{-1}_n(Q \, r) \quad \text{with } r \in \mathbb{C}^q$$

where vec$_n^{-1}(x)$ puts the vector $x$ into an $n$-row matrix $X$ so that $x = \text{vec}(X)$. This expression completely characterizes the set of matrices $R$ regardless of the properties of the given matrix $A$.

In the present paper, we propose a numerical method to compute a matrix $Q$ representing a basis for the null space $Q$. This is a quick and reliable procedure that takes advantage of the Schur decomposition of the matrix $A$ (see, e.g., Horn et al. [2]).

The structure of the paper is as follows. In Section 2, we give characteristics of the null space of the matrix $(A^T \otimes I_n - I_n \otimes A)$ relying on Kronecker product properties. In Section 3, we end up with an algorithm for evaluating a basis for the null space of this matrix. The procedure is based on the Schur decomposition of the matrix $A$. Finally, in Section 4, we illustrate the numerical complexity of the proposed algorithm and compare it that other possible ways of obtaining a basis of the mentioned null space.

## 2 Basis for the null space $Q$

A first information about the null space $Q$ concerns its dimension, denoted $q$. By definition, it is identical to the geometric multiplicity [3] of the zero eigenvalue of the underlying matrix $(A^T \otimes I_n - I_n \otimes A)$. Therefore, let us recall a basic result [2, Theo. 4.4.5] concerning Sylvester equation.

**Theorem 1** Let the matrices $A \in \mathbb{C}^{n \times n}$ and $B \in \mathbb{C}^{m \times m}$ have eigenvalue sets $\sigma(A) = \{\lambda_1, \ldots, \lambda_n\}$ and $\sigma(B) = \{\mu_1, \ldots, \mu_m\}$, respectively. Then,

$$\sigma(I_m \otimes A + B \otimes I_n) = \{\lambda_i + \mu_j; \; 0 < i \leq n, \; 0 < j \leq m\}$$
Furthermore, if \( x_i \in \mathbb{C}^n \) and \( y_j \in \mathbb{C}^m \) are eigenvectors associated to \( \lambda_i(A) \) and \( \mu_j(B) \), respectively, then the vector \( y_j \otimes x_i \in \mathbb{C}^{mn} \) is an eigenvector of \((I_m \otimes A + B \otimes I_n)\) corresponding to the eigenvalue \( \lambda_i + \mu_j \).

By applying this result to the matrix \((A^T \otimes I_n - I_n \otimes A)\), i.e. by putting \( B = -A^T \), it yields the dimension of \( Q \) as

\[
q = \sum_{i=1}^{n} m_g(\lambda_i)
\]

where \( m_g(\lambda) \) stands for the geometric multiplicity of the eigenvalue \( \lambda \). Note that, as the summation index \( i \) passes by each of the eigenvalue indices, the geometric multiplicity of identical eigenvalues contribute (to \( q \)) as many times as the value of their algebraic multiplicity.

One can think to use the result of Theorem 1 in order to provide a basis for the \( q \)-dimensional subspace \( Q \). In fact, it is relevant only in situations where the algebraic and the geometric multiplicity of the eigenvalues of \( A \) are identical. Indeed, the following result holds.

**Lemma 2** Let \((\lambda_i, x_i)\) and \((\lambda_i, y_i)\) be the \( i \)-th eigenpairs of the matrices \( A \) and \( A^T \) (for \( 0 < i \leq n \)), respectively. Then, the dimension of the subspace spanned by the set \( \{y_i \otimes x_j \text{ such that } \lambda_i = \lambda_j; 0 < i, j \leq n\} \) is identical to

\[
\sum_{i=1}^{n} \frac{m_g^2(\lambda_i)}{m_a(\lambda_i)}
\]

where \( m_a(\lambda) \) stands for the algebraic multiplicity of the eigenvalue \( \lambda \).

**Proof:** First, each distinct eigenvalue \( \lambda \) of the matrix \( A \) (resp. \( A^T \)) is associated to \( m_g(\lambda) \) independent eigenvectors, i.e. \( x_i \) (resp. \( y_i \)) for \( 0 < j \leq n \) subject to \( \lambda_i = \lambda \). By combining these eigenvectors, the total number of independent Kronecker product vectors, i.e. \( y_j \otimes x_i \), that such eigenvalue leads to is exactly \( m_g^2(\lambda) \).

Then, by taking the contribution of the distinct eigenvalues into account, the dimension of the space spanned by the associated Kronecker product vectors (that form the set mentioned in the lemma) is identical to \( \sum_i m_g^2(\lambda_i) \) where the sum only take cares of distinct \( \lambda_i \)'s.

Finally, the proof is completed if we let the summation index \( i \) go over all the eigenvalue indices and compensate for it by dividing by the number of time an identical eigenvalue, e.g. \( \lambda_i \), contributes to the summation, i.e. its algebraic multiplicity, e.g., \( m_a(\lambda_i) \).

The set mentioned in this lemma takes account of all the eigenvectors corresponding to zero eigenvalues of \((A^T \otimes I_n - I_n \otimes A)\) (see Theorem 1). Note also that the vectors \( x_i \) and \( y_i \) are the \( i \) column of the matrices \( X = V \) and \( Y = V^{-T} \) (for nonsingular \( V \)), respectively.

It clearly appears that the dimension of the subspace spanned by this set coincides with
that of $Q$ given in expression (4) only when $m_q(\lambda_i) = m_a(\lambda_i)$ for all $i$, i.e. diagonalizable matrix $A$.

In a simpler situation where all $\lambda_i$'s are distinct, it can be shown that the expressions (1) and (3) are similar. This is derived as follows. First, the $i$-th column of the matrix $Q$ can be expressed in terms of the corresponding columns in the matrices $V$ and $V^{-T}$ as $Q_i = y_i \otimes x_i$ with $0 < i \leq n$. Then, by taking care of eigenvectors corresponding to nonzero eigenvalues of $(A^T \otimes I_n - I_n \otimes A)$, the expression (3) becomes

$$R = \text{vec}_n^{-1} \left( (V^{-T} \otimes V) \tilde{r} \right)$$

where $\tilde{r} = \text{vec}(\text{diag}(r))$ with $r \in \mathbb{C}^n$. Hence, it leads to expression (1) after extracting the “vec”-inverse operator.

In case of non-distinct eigenvalues having identical algebraic and geometric multiplicities, i.e. $m_q(\lambda_i) = m_a(\lambda_i)$ for all $i$'s, the only difference with this expression is that the $n^2$-dimensional vector $\tilde{r}$ is not anymore identical to the vectorization of a diagonal matrix but to that of a block diagonal matrix having non-diagonal nonzero entries for indices corresponding to identical eigenvalues.

As was seen above, in the case of nondiagonalizable matrix $A$, the set mentioned in Lemma 2 fails in providing a basis for the null space $Q$. The reason for this is simply that this set is no more a basis: it has dependent vectors because of the singularity of the eigenvector matrices $X = V$ and $Y$ (for $A^T$). So, the $q$ columns of the matrix $(Y \otimes X)$ corresponding to the zero eigenvalues of $(A^T \otimes I_n - I_n \otimes A)$ do not form a full-column rank matrix.

This is of course bad news because the matrix $Q$ in expression (3) can not be constructed by use of the matrices $X$ and $Y$ anymore. Instead, it must be referred to its prime definition of being the matrix representation of a basis of the null space of the matrix $(A^T \otimes I_n - I_n \otimes A)$.

With this in mind, a first method to end up with a matrix $Q$ is to perform the singular value decomposition (SVD) of $(A^T \otimes I_n - I_n \otimes A)$ and take up the last $q$ columns of the right singular vector matrix. Unfortunately, its numerical cost is far too prohibitive as no care is taken of the structure brought by the Kronecker product.

More attractive is the method we propose in the next section. It is a numerical procedure that iteratively computes the columns of a particular matrix $Q$. It is based on the Schur decomposition of the matrix $A$ and takes advantages of the matrix structure. Furthermore, it is low cost and numerically robust.

3 Construction of a null space basis

In this section, we present a procedure that ends up with a particular matrix $Q$. The fundamental tool for it is the Schur decomposition of the matrix $A$. It leads to

$$A = U T_n U^*$$

4
where $T_u \in \mathbb{C}^{n \times n}$ is upper-triangular and $U \in \mathbb{C}^{n \times n}$ is unitary while the symbol $*$ stands for the conjugate-transpose operator. Note that the diagonal elements of $T_u$ are the eigenvalues of $A$, i.e. $T_{u,ii} = \lambda_i$ say.

By use of this representation for $A$, the commutation condition (2) becomes

$$(T_u^T \otimes I_n - I_n \otimes T_u) (U^T \otimes U^*) \text{ vec}(R) = 0$$

Thus, if $P$ denotes the matrix representation of a basis of the null space of the matrix

$$(T_u^T \otimes I_n - I_n \otimes T_u)$$

then, from standard Kronecker product manipulations, a possible matrix $Q$ is $(U^* \otimes U) P$. This readily implies that $P \in \mathbb{C}^{n^2 \times q}$ so that the two null spaces have the same dimension, i.e. $q = \text{dim}(Q)$. Note that the columns of $Q$ are simply evaluated as $Q_m = \text{vec}(U \text{vec}^{-1}(P_m U^*))$ for $0 < m \leq q$.

Furthermore, the transformation is unitarily invariant: $P^* P = I_q \Leftrightarrow Q^* Q = I_q$.

Now, let us construct a particular matrix $P$. Therefore, we first enlighten the structure of the matrix $(T_u^T \otimes I_n - I_n \otimes T_u)$. It is a lower block diagonal matrix in which the diagonal blocks are upper-triangular while each of the other nonzero blocks is a diagonal depending on one element only.

For what concerns the null space of this matrix, it has a dimension that is the sum of the dimension of the null space of each diagonal blocks, i.e.

$$q = \sum_{i=1}^{n} q_i \quad \text{with} \quad q_i = m_g(\lambda_i)$$

This is exactly the result expressed in (4). More importantly, this fact provides us a hint to iteratively evaluate the matrix $P$. This is done by considering the null space of each diagonal block separately.

Let us detail the procedure for the $i$-th diagonal block, i.e. $(T_{u,ii}I_n - T_u)$.

First, assume that its null space is spanned by the set $\{v_{i,k}; 0 < k \leq m_g(\lambda_i)\}$, e.g. coming from the last $m_g(\lambda_i)$ right singular vectors $v_{i,k}$ of the SVD of this block. To each vector $v_{i,k}$ corresponds a column in the matrix $P$. More precisely, denote by $P_m$ the column associated to $v_{i,k}$, i.e. $m = m(i, k)$ because $k \in [1, m_g(\lambda_i)]$ for $i \in [1, n]$ while $m \in [1, q]$.

State also that

$$P_m = (0, \ldots, 0, v_{i,k}^T, [P_m]_{i+1}^T, \ldots, [P_m]_n^T)^T$$

where $[x]_i \in \mathbb{C}^n$ stands for a sub-vector of $x \in \mathbb{C}^{n^2}$ according to $x = ([x]_1^T, \ldots, [x]_n^T)^T$.

Then, it remains to impose that $P_m$ belongs to the null space of $(T_u^T \otimes I_n - I_n \otimes T_u)$. This is done by recursively solving linear sets of equations, i.e.

$$(T_u - T_{u,ji}I_n) |P_m|_j = \sum_{l=i}^{j-1} T_{u,li} |P_m|_l$$

for $j$ increasing from $i + 1$ to $n$. Note that such equation sets also appear in the resolution of the Sylvester equation (see, e.g., Golub et al. [1, chap. 7]).
Thus, the resulting column $P_m$ fulfills the null space requirements. For each $k \in [1, m_g(\lambda_i)]$, we repeat the statement and the equation set resolution for the associated $P_{m(i,k)}$.

Finally, the complete matrix $P$ is obtained after performing this procedure for all the diagonal blocks, i.e. $(T_u,iiI_n - T_u)$ for $0 < i \leq n$. Note that, once the matrix $P$ is known, it is easy to evaluate an unitary version of it by use of Gram-Schmidt orthogonalization [3].

In summary, we end up with the algorithm described in Figure 1 where the symbol “†” stands for the pseudo-inverse operator [3, chap. 3].

\begin{center}
\begin{tabular}{|c|}
\hline
Schur decomposition of $A \in \mathbb{C}^{n \times n}$ giving $A = U^* T_u U$ \\
\hline
For $i = n : -1 : 1$
\begin{itemize}
\item SVD$(T_u,iiI_n - T_u)$ so that $\{v_{i,k}; 0 < k \leq m_g(\lambda_i)\}$ and $Z_i = (T_u - T_u,ijI_n)^\dagger$
\item For $k = 1 : m_g(\lambda_i)$
\begin{itemize}
\item Let $P_m = (0, \ldots, 0, v_{i,k}^T, [P_m]^T_{i+1}, \ldots, [P_m]^T_{n})^T$ with $m = m(i, k)$
\item For $j = i + 1 : n$
\begin{equation*}
[P_m]_j = Z_j \sum_{l=i}^{j-1} T_u,ij [P_m]_l
\end{equation*}
\end{itemize}
\end{itemize}
End loop on $j$, $k$ and $i$
\end{tabular}
\end{center}

Figure 1: Algorithm for the construction of the matrix $P$

Let us make the following remarks.

- At each step $i$, the SVD of $(T_u,iiI_n - T_u)$ is computed so that the dimension and a basis of its null space are robustly evaluated. This makes the whole algorithm numerically reliable for the estimation of the matrix $P$. Furthermore, the pseudo-inverse of this diagonal block matrix is computed at this step and is memorized (in $Z_i$) for being used $(n - i)$ times in the $j$ steps.

- The algorithm complexity is as follows. There are $n$ SVD’s and $n(n - 1)/2$ matrix-vector multiplication so that the number of flops is of order $n^4$. This is to be compared with the brute-force null space estimation of the original matrix $(A^T \otimes I_n - I_n \otimes A)$ whose complexity is of order $n^6$ because it is based on the SVD of this $n^2$ matrix.

Finally, note that the proposed algorithm has treated the problem of finding the null space of the matrix $(A^T \otimes I_n - I_n \otimes A)$ regardless of the properties of the underlying matrix $A$. By that we mean that it has not been taken care of possibilities for this latter matrix to be block-diagonalizable for which a part of the desired null space could have
been expressed in terms of the Kronecker product of left and right eigenvectors of the matrix $A$, i.e. those associated to eigenvalues having identical geometric and algebraic multiplicities. We leave taylorized resolutions of such situations to the reader.

4 Numerical discussions

In this section, we intend to compare the numerical complexity of the computation of the null space of the matrix $(A^T \otimes I_n - I_n \otimes A)$ by use of three different methods: namely,

1. the algorithm presented in Figure 1 for evaluating the matrix $P$ followed by the formula for ending up with a non-unitary matrix $Q$, i.e. $Q_m = \text{vec}(U \text{vec}^{-1}(P_m)U^*)$ for $0 < m \leq q$.

2. while considering that the given matrix $A$ is diagonalizable with eigenvalues having identical geometric and algebraic multiplicities, the columns of the matrix $Q$ evaluated as the Kronecker product of left and right eigenvectors of the matrix $A$. The resulting matrix is not unitary.

3. the SVD of $(A^T \otimes I_n - I_n \otimes A)$ so that the columns of the matrix $Q$ are the right singular vectors corresponding to the zero singular values. This matrix is unitary.

Obviously, these three methods have different numerically complexities. Moreover, we also evaluate the complexity of the Gram-Schmidt orthogonalization procedure.

The simulations are performed on a Sun Ultra 1/170E with 128 MB RAM by use of the Matlab 5.1. The results are presented in two different graphs presented in Figure 2. They illustrate the number of “flops” (as given by the Matlab function `flops`) as well as the time (in sec.) needed for evaluating the matrix $Q$ by use of these three methods as a function of the dimension $n$ of the matrices $A$ which have been taken randomly. In the left-hand side graph, we have also drawn the number of “flops” related to the Gram-Schmidt orthogonalization of the matrix $Q$.

From the left-hand side figure, we can evaluate approximated complexities for the three methods as well as for the Gram-Schmidt procedure. The result is shown in Table 1. As expected, the cost of the third method is far too prohibitive, i.e. $\sim 12 n^2/80$ times the complexity of that proposed in the paper. When comparing the numerical cost of the first two methods after Gram-Schmidt orthogonalization of the matrix $Q$, we obtain

$$\frac{\mathcal{g}(1)}{\mathcal{g}(2)} \approx \frac{40}{n + 40}$$

Hence, it is similar to $n$ for small $n$ but always less than 40 whatever $n$.

As far as the time is concerned, it appears that only the third procedure is crucially too slow while the two others spend less than one second for a matrix $A$ of dimension less than $n = 20$. The quantity $12 n^2/82$ that is the ratio between the complexity of the third and of the first procedure really makes the difference in time: 100 seconds at $n = 25$!
Figure 2: Number of “flops” (left) taken and time (right) spent for evaluating a basis for the null space $Q$ by use of the three methods (in ‘—’, ‘—’ and ‘—’, respectively). The number of “flops” needed for the Gram-Schmidt orthogonalization of the resulting matrix $Q$ (in ‘—’).

<table>
<thead>
<tr>
<th>Method</th>
<th>$\sim z$ (flops)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$80 n^4$</td>
</tr>
<tr>
<td>2</td>
<td>$80 n^3$</td>
</tr>
<tr>
<td>3</td>
<td>$12 n^6$</td>
</tr>
<tr>
<td>GS</td>
<td>$2 n^3(n+1)$</td>
</tr>
</tbody>
</table>

Table 1: Approximated complexity of the three methods and of the Gram-Schmidt procedure.

5 Conclusion

In this paper, we have studied the commuting characterization of matrices with a given matrix $A \in \mathbb{C}^{n \times n}$. It has appeared that it is related to the null space of a well-structured matrix, i.e. $(A^T \otimes I_n - I_n \otimes A)$. More precisely, any matrix $R$ that commutes with $A$ has its vectorized column in the null space of that matrix. Hence, the problem of finding such matrix $R$ is equivalent to that of obtaining a representation of a basis for that space.

From a numerical point of view, we have proposed a low cost and robust procedure for computing a basis for the “commuting” null space. This algorithm is based on the Schur decomposition of the matrix $A$. Finally, we have provided an analysis of its numerical complexity that is of order $80 n^4$. 
References

