SIAM J. Matrix Anal. Appl.
Vol. 36, No. 1, pp. 38–54

AN EFFICIENT ALGORITHM FOR COMPUTING THE GENERALIZED NULL SPACE DECOMPOSITION[*]

NICOLA GUGLIELMI[†], MICHAEL L. OVERTON[‡], AND G. W. STEWART[§]

Abstract. The generalized null space decomposition (GNSD) is a unitary reduction of a general matrix A of order n to a block upper triangular form that reveals the structure of the Jordan blocks of A corresponding to a zero eigenvalue. The reduction was introduced by Kublanovskaya. It was extended first by Ruhe and then by Golub and Wilkinson, who based the reduction on the singular value decomposition. Unfortunately, if A has large Jordan blocks, the complexity of these algorithms can approach the order of n^4. This paper presents an alternative algorithm, based on repeated updates of a QR decomposition of A, that is guaranteed to be of order n^3. Numerical experiments confirm the stability of this algorithm, which turns out to produce essentially the same form as that of Golub and Wilkinson. The effect of errors in A on the ability to recover the original structure is investigated empirically. Several applications are discussed, including the computation of the Drazin inverse.

Key words. generalized null space, Jordan form, staircase algorithms, Drazin inverse

AMS subject classifications. 65F15, 65F30

DOI. 10.1137/140956737

1. Introduction.

The primary purpose of this paper is to present a new algorithm for computing the following decomposition, which we call the generalized null space decomposition (GNSD). Let A be a matrix of order n and let ν be the index of A; i.e., the smallest integer j for which the null spaces $\mathcal{N}(A^j)$ and $\mathcal{N}(A^{j+1})$ are equal. Then there is a unitary matrix V such that

$$V^*AV = B = \begin{pmatrix} 0 & B_{12} & B_{13} & \cdots & B_{1,\nu-1} & B_{1,\nu} & B_{1,\nu+1} \\ 0 & 0 & B_{23} & \cdots & B_{2,\nu-1} & B_{2,\nu} & B_{2,\nu+1} \\ 0 & 0 & 0 & \cdots & B_{3,\nu-1} & B_{3,\nu} & B_{3,\nu+1} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & B_{\nu-1,\nu} & B_{\nu-1,\nu+1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & B_{\nu,\nu+1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & B_{\nu+1,\nu+1} \end{pmatrix}, \tag{1.1}$$

where we have the following:

(1.2)
1. The diagonal blocks of B are square.
2. The superdiagonal blocks $B_{12}, \ldots, B_{\nu-1,\nu}$ are of full column rank.
3. The block $B_{\nu+1,\nu+1}$ is nonsingular (provided it is not of order 0).

[*]Received by the editors February 12, 2014; accepted for publication (in revised form) by D. P. O'Leary October 22, 2014; published electronically January 22, 2015.
http://www.siam.org/journals/simax/36-1/95673.html

[†]Dipartimento di Ingegneria Scienze Informatiche e Matematica, Università dell'Aquila, Italy (guglielm@units.it, http://www.univaq.it/∼guglielm). This author's research was supported in part by the Italian Istituto Nazionale di Alta Matematica, Gruppo Nazionale di Calcolo Scientifico (I.N.d.A.M., G.N.C.S.).

[‡]Courant Institute of Mathematical Sciences, New York University, New York, NY 10012 (overton@cs.nyu.edu, http://www.cs.nyu.edu/∼overton). This author's research was supported in part by National Science Foundation Grant DMS-1317205.

[§]Department of Computer Science, University of Maryland, College Park, MD 20742 (stewart@cs.umd.edu, http://www.cs.umd.edu/∼stewart).

The importance of this decomposition lies in its relation to the Jordan blocks corresponding to a zero eigenvalue of A, which we will call the *zero Jordan blocks* and will denote by $J_k(0)$, where k is the order of the block. Specifically, it can be shown that if

$$V = (V_1 \ V_2 \ \cdots \ V_{\nu+1})$$

is partitioned conformally with B, then for $j = 1, \dots, \nu$

$$\mathcal{V}_j = \operatorname{span}(V_j) = \operatorname{span}(\{x\colon A^{j-1}x \neq 0 \text{ and } A^j x = 0\}). \tag{1.3}$$

(For a proof of this fact, see Appendix C.) Now, for $j = 1, \dots, \nu$ let μ_j be the order of the jth diagonal block of B and let $\mu_{\nu+1} = 0$. Note that from the second item in (1.2) it follows that $\mu_j \geq \mu_{j+1}$. By inspecting the zero Jordan blocks of powers of A it is easy to see that

$$\mu_j \text{ is the number of zero Jordan blocks of order} \geq j. \tag{1.4}$$

Hence,

$$\mu_j - \mu_{j+1} \text{ is the number of zero Jordan blocks of order } j.$$

Thus the dimensions of the diagonal blocks of the GNSD of A provide a complete description of the zero Jordan blocks of A.

In the terminology of Jordan canonical forms the nonzero members of $\mathcal{V}_j$ $(j \leq \nu)$ are generalized eigenvectors of grade j. Since in this case the eigenvalue in question is zero, we will call them *null vectors of grade* j and call $\mathcal{V}_j$ the *generalized null space of grade* j. The union of these null spaces is simply called the generalized null space.

Given the above connections, it is not surprising that the GNSD arose in attempts to compute the Jordan canonical form of a matrix. In 1966 Kublanovskaya [16] introduced the decomposition, and variants appeared in papers by Ruhe [17] in 1970, by Golub and Wilkinson [11] in 1976, and, together with accompanying software, by Kågström and Ruhe [15]. The form given here is essentially the one given by Golub and Wilkinson. Van Dooren [20] extended this work on the Jordan form to the Kronecker canonical form for a matrix pencil, introducing the term "staircase form" to indicate the resemblance of the transpose of the right-hand side of (1.1) to a descending flight of stairs. For a survey on the subsequent history of "staircase algorithms," see [10, sect. 1.1]. In this paper we will drop the staircase nomenclature, and refer to Ruhe's and Golub and Wilkinson's algorithms as the SVD algorithm for computing the GNSD. For reasons that will become clear later, we will call our algorithm the QR/update algorithm.

The GNSD can be computed as follows. Let $B^{(1)} = A$. If $B^{(1)}$ is nonsingular, set $B = B^{(1)}$ and $V = I$. If not, let $V_1^{(1)}$ be an orthonormal basis for $\mathcal{N}(B^{(1)})$ and let $V^{(1)} = (V_1^{(1)} \ V_2^{(1)})$ be unitary. Then

$$B^{(2)} = V^{(1)*} B^{(1)} V^{(1)} = \begin{pmatrix} 0 & B_{12}^{(2)} \\ 0 & B_{22}^{(2)} \end{pmatrix}. \tag{1.5}$$

If $B_{22}^{(2)}$ is nonsingular, the process ends with $V = V^{(1)}$ and $B = B^{(2)}$. Otherwise the process is repeated recursively with the matrix $B_{22}^{(2)}$, until there is nothing left to reduce. At all stages the orthogonal transformations are accumulated to form the matrix V.[1]

All that is lacking to turn the above sketch into a working algorithm is a method for determining bases for the null space of the matrices $B_{kk}^{(k)}$. Kublanovskaya used a pivoted QR decomposition, while first Ruhe and then Golub and Wilkinson used the more reliable singular value decomposition. Either alternative requires $O(n_k^3)$ operations, where n_k is the order of $B_{kk}^{(k)}$. Since the algorithm requires that ν such decompositions be computed, if ν is small the process of obtaining null vectors is $O(n^3)$. As ν approaches n, however, the work required approaches $O(n^4)$, which is prohibitive for large n.[2]

The main contribution of this paper is to describe an $O(n^3)$ algorithm for computing the GNSD. The basic idea is to deflate individual generalized null vectors one at a time. The approximate null vectors are obtained from a QR factorization of A by a process that goes under the rubric of condition estimation (see [14, Chap. 15]). The QR factorization is updated as the null vectors are deflated. All this is described in the next section along with how the process can be implemented using plane rotations. The process is numerically stable, in the sense that the computed B satisfies $VBV^* = A + E$, where E is of the order of the rounding unit plus the errors introduced by the use of approximate null vectors. The latter errors also occur with the SVD algorithm.

A related precursor that also works with a QR factorization has been sketched by Beelen and Van Dooren [3]. However, they use Tony Chan's rank revealing QR decomposition [5] to determine the generalized null spaces. Unfortunately, this algorithm is, in theory, unreliable except for subspaces of small dimension. They then deflate the subspace from the QR decomposition. Implementation is not discussed.

Two other $O(n^3)$ algorithms have appeared in the literature. The first is again due to Golub and Wilkinson, and like the QR/update method it produces generalized null vectors sequentially [11, sect. 11]. However, these vectors are not orthogonal and may become nearly linearly dependent. The second algorithm, due to Anstreicher and Rothblum [1], uses a variant of the Gauss–Jordan elimination to compute generalized null vectors. The authors do not comment on the numerical aspects of their algorithm. But there is no guarantee of the numerical independence of the vectors, and the Gauss–Jordan algorithm itself has instabilities. (For more on the Gauss–Jordan algorithm, see [14, pp. 273–277].)

Up to now we have ignored the problem of errors. In any approach to computing the GNSD that involves errors, the user must furnish a tolerance to determine when a computed singular value is to be considered zero or when an approximate null vector is satisfactory. The result is to introduce errors into the decomposition that may be as large as the order of the tolerance. Nonetheless, the computed GNSD can be shown to be the exact decomposition of a perturbation of A whose norm can be explicitly computed.

Unfortunately, this property is not as useful as it appears. For if A is perturbed

[1]This sketch does not establish item 2 of (1.2). However, it is easily seen that if B_{12} in (1.1) is not of full column rank, then $B^{(1)}$ has an additional null vector, contradicting the fact that a full complement of null vectors was used in the first step. Similar considerations hold for the remaining $B_{k,k+1}$.

[2]The cost accumulating the orthogonal transformations may also be prohibitive if performed naively. However, if the $V_k^{(k)}$ are expressed as products of Householder transformations, the cost is manageable.

by rounding error, the result will, in general, be nonsingular and ν will be zero. Thus the proper question to ask about any algorithm for computing a GNSD is whether it can recover the generalized null space structure from the perturbed matrix. There is, in fact, good reason to believe that no algorithm can do so in all cases. Hence, in section 3, we introduce a class of test matrices—essentially matrices obtained from a Jordan form by a similarity transformation of known condition number κ. For this class there are reasonable choices for the tolerance mentioned above. However, we were surprised to find that the ability to recover the original Jordan structure may depend on the square of the condition number of the transformation. This κ^2 effect limits the problems we can solve, but at least some tractable problems remain.

The numerical experiments of section 4 confirm the numerical stability of the QR/update algorithm. They also show that the QR/update and SVD algorithms compute the same decomposition, even when they fail to recover the original Jordan structure. Unfortunately, they also confirm the reality of the κ^2 effect. A summary of the paper is given in section 5.

Appendix A shows how the GNSD of a matrix A can be used to compute its Drazin generalized inverse. The advantage of this algorithm is that, aside from orthogonal transformations, it consists of solving a single Sylvester equation, which can be done efficiently using the output of the QR/update algorithm. Appendix A also provides a new derivation of a formula of Hartwig relating the Drazin inverse to an arbitrary generalized inverse. Appendix B describes two applications in which we have found the GNSD to be useful. Finally, in Appendix C we establish the characterization (1.3).

Throughout this paper $\|\cdot\|$ will denote the Euclidean vector norm and the spectral matrix norm.

2. The QR/update algorithm.

The algorithm proposed in this paper begins with a QR factorization $A = B^{(1)} = Q^{(1)}R^{(1)}$ and deflates null vectors of $R^{(1)}$ to produce a QR factorization $B_{22}^{(2)} = Q^{(2)}R^{(2)}$, where, up to a unitary similarity, $B_{22}^{(2)}$ is the matrix appearing in (1.5). The process continues by producing successive matrices $B_{kk}^{(k)}$ until after ν steps $B_{\nu+1,\nu+1}^{(\nu+1)}$ has no null vectors.

We will now describe the first stage of this algorithm in detail. In this description we will assume that the computations are exact and that null vectors, however computed, are exact. To avoid notational clutter, we will drop subscripts and superscripts and use math accents to indicate the various substeps.

The process starts with the QR factorization $B = QR$. Let x be a normalized null vector of R (if there is one). Let $\bar{V}$ be a unitary matrix such that $\bar{V}^*x = \mathbf{e}_1$ and let U be a unitary matrix such that $\hat{R} = U^*R\bar{V}$ is upper triangular. Set $\hat{Q} = \bar{V}^*QU$ so that

$$\hat{B} = \bar{V}^*B\bar{V} = \hat{Q}\hat{R}.$$

Now $\hat{R}\mathbf{e}_1 = (U^*R\bar{V})(\bar{V}^*x) = U^*Rx = 0$. Hence, the first column of $\hat{R}$ is zero.

Note that the final value of V may be obtained by initializing V to the identity matrix and, as each matrix $\bar{V}$ is generated, replacing V by $V\bar{V}$. Likewise, B must be initialized to A and for each $\bar{V}$ replaced by $\bar{V}^*B\bar{V}$.

Next determine a unitary matrix W such that $\tilde{R} = W^*\hat{R}$ is upper triangular with its first row zero. Set $\tilde{Q} = \hat{Q}W$ and partition $\tilde{Q} = (\tilde{q}_1\ \tilde{Q}_2)$. Then[3]

$$\hat{B} = \hat{Q}\hat{R} = (\hat{Q}W)(W^*\hat{R}) = \tilde{Q}\tilde{R} = (\tilde{q}_1\ \tilde{Q}_2)\begin{pmatrix} 0 & 0 \\ 0 & \tilde{R}_{22} \end{pmatrix} \equiv \tilde{B}. \tag{2.1}$$

The vector $\mathbf{e}_1$ is a null vector of $\tilde{B}$. If $\tilde{R}_{22}$ is singular, the process can be continued as above by computing a null vector of $\tilde{R}_{22}$ and using it to zero its first row and column. This process continues until, after say k steps, the resulting triangular matrix—call it $\breve{R}_{22}$—is nonsingular. At this point B has been transformed into

$$\breve{B} = (\breve{Q}_1\ \breve{Q}_2)\begin{pmatrix} 0 & 0 \\ 0 & \breve{R}_{22} \end{pmatrix}, \tag{2.2}$$

where $\breve{Q}_1$ has k columns. It follows that $\breve{B}$ has the form

$$\breve{B} = \begin{pmatrix} 0 & \breve{B}_{12} \\ 0 & \breve{B}_{22} \end{pmatrix}.$$

This completes a single step of the general procedure described above. As noted previously, up to a unitary similarity $\breve{B}$ is just the matrix $B_{22}^{(2)}$ in (1.5).

To continue the algorithm, a QR decomposition of $\breve{B}_{22} = B_{22}^{(2)}$ is needed. Fortunately, it does not have to be computed from scratch. Specifically, the QR decomposition in (2.2) can be truncated to give the QR factorization

$$\begin{pmatrix} \breve{B}_{12} \\ \breve{B}_{22} \end{pmatrix} = \breve{Q}_2\breve{R}_{22}. \tag{2.3}$$

Thus all that is required is to remove $\breve{B}_{12}$ from this factorization. It turns out there are efficient, stable algorithms to perform this downdating (see [8, 21]). For later reference when we come to operation counts, note that these algorithms remove $\breve{B}_{12}$ a row at a time with a cost of $O(n^2)$ for each row.

Although we will discuss the effects of errors and perturbations on the algorithm later, the following problem is best treated here. Owing to errors in the original matrix A, the vector Ax will not be exactly zero. Suppose we decide to accept x as an approximate null vector if it satisfies $\|Ax\| \le \tau$. Then the decomposition $\tilde{B}$ in (2.1) will be altered to a matrix $\ddot{B}$ of the form

$$\ddot{B} = \begin{pmatrix} \pi & 0 \\ p & \ddot{R}_{22} \end{pmatrix},$$

where $\|(\pi\ p^*)\| \le \tau$.

Now suppose that the original R has μ_1 singular values, $\sigma_n, \ldots, \sigma_{n-\mu_1+1}$, that are less than τ (recall that μ_1 is the dimension of the null space of A). Is it possible for $\ddot{R}_{22}$ to have fewer than $\mu_1 - 1$ singular values less than τ? In such a case the algorithm would return a GNSD with its first diagonal block of order less than the required order μ_1. To see that this cannot happen note that the singular values of $\ddot{B}$ are the same as those of the original R. Now consider the matrix

$$\ddot{B}^*\ddot{B} = \begin{pmatrix} \pi^2 + p^*p & p^*\ddot{R}_{22} \\ \ddot{R}_{22}^*p & \ddot{R}_{22}^*\ddot{R}_{22} \end{pmatrix}.$$

[3]Although $\hat{B}$ and $\tilde{B}$ are identical, they are associated with different QR factorizations; i.e., the first row of $\tilde{R}$ is zero, whereas the first row of $\hat{R}$ is, in general, nonzero. It is worth noting that this is only possible because $\hat{R}$ is singular.

By the interlacing theorem [19, Thm. IV.4.2], the smallest $\mu_1 - 1$ eigenvalues of $\ddot{R}_{22}^*\ddot{R}_{22}$—call them $\ddot{\sigma}_n^2, \ldots, \ddot{\sigma}_{n-\mu_1+2}$—satisfy

$$\sigma_n^2 \le \ddot{\sigma}_n^2 \le \sigma_{n-1}^2 \le \ddot{\sigma}_{n-1}^2 \le \cdots \le \ddot{\sigma}_{n-\mu_1+2}^2 \le \sigma_{n-\mu_1+1}^2 < \tau^2.$$

It follows that $\ddot{R}_{22}$ has at least $\mu_j - 1$ singular values less than τ.[4]

As mentioned above, our algorithm can be implemented by plane rotations (see [18, sect. 4.1.3] for details on how to manipulate these transformations). The first job is to determine $\bar{V}$ so that $\bar{V}^*x = \mathbf{e}_1$. Suppose that $n = 4$. Let V_{34}^* be a rotation in the $(3,4)$-plane that zeros the last element of x. In terms of Wilkinson diagrams,

$$V_{34}^* \begin{pmatrix} x \\ x \\ x \\ x \end{pmatrix} = \begin{pmatrix} x \\ x \\ x \\ 0 \end{pmatrix}.$$

Next, let V_{23}^* be a rotation in the $(2,3)$-plane that zeros the third element of V_{34}^*x:

$$V_{23}^* \begin{pmatrix} x \\ x \\ x \\ 0 \end{pmatrix} = \begin{pmatrix} x \\ x \\ 0 \\ 0 \end{pmatrix}.$$

Finally, let V_{12}^* be a plane rotation in the $(1,2)$-plane that zeros the second element of $V_{23}^*V_{34}^*x$:

$$V_{12}^* \begin{pmatrix} x \\ x \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} x \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Then $\bar{V} = V_{34}V_{23}V_{12}$. As noted above, the rotations that generate $\bar{V}$ must be accumulated in V and used to update B.

The next step is to compute the matrix U. This is done at the same time as $\bar{V}$ is postmultiplied into R. Specifically, the matrix RV_{34} has a subdiagonal element in the $(4,3)$ position:

$$RV_{34} = \begin{pmatrix} r & r & r & r \\ 0 & r & r & r \\ 0 & 0 & r & r \\ 0 & 0 & r & r \end{pmatrix}.$$

This element can be eliminated by a rotation U_{34}^* to give

$$U_{34}^*RV_{34} = \begin{pmatrix} r & r & r & r \\ 0 & r & r & r \\ 0 & 0 & r & r \\ 0 & 0 & 0 & r \end{pmatrix}.$$

[4]It may be asked if it is possible for $\ddot{R}_{22}$ to have too *many* singular values less than τ. The answer is yes, but only if $\sigma_{n-\mu_j}$ is near τ, in which case the entire decompostion is ill determined.

The next step is to process V_{23}. As above,

$$U_{34}^* R V_{34} V_{23} = \begin{pmatrix} r & r & r & r \\ 0 & r & r & r \\ 0 & r & r & r \\ 0 & 0 & 0 & r \end{pmatrix}, \qquad U_{23}^* U_{34}^* R V_{34} V_{23} = \begin{pmatrix} r & r & r & r \\ 0 & r & r & r \\ 0 & 0 & r & r \\ 0 & 0 & 0 & r \end{pmatrix}.$$

Finally,

$$U_{23}^* U_{34}^* R V_{34} V_{23} V_{12} = \begin{pmatrix} r & r & r & r \\ r & r & r & r \\ 0 & 0 & r & r \\ 0 & 0 & 0 & r \end{pmatrix},$$

and

$$U_{12}^* U_{23}^* U_{34}^* R V_{34} V_{23} V_{12} = \hat{R} = \begin{pmatrix} r & r & r & r \\ 0 & r & r & r \\ 0 & 0 & r & r \\ 0 & 0 & 0 & r \end{pmatrix}.$$

As argued above, the $(1,1)$-element of $\hat{R}$ must be zero. The matrix $U = U_{34}U_{23}U_{12}$, and the rotations may be accumulated in Q to form $\hat{Q}$.

To generate W let W_{12} be a plane rotation that zeros the $(1,2)$-element of $\hat{R}$:

$$W_{12}^* \hat{R} = \begin{pmatrix} 0 & 0 & r & r \\ 0 & r & r & r \\ 0 & 0 & r & r \\ 0 & 0 & 0 & r \end{pmatrix}.$$

The next two steps proceed similarly:

$$W_{13}^* W_{12}^* \hat{R} = \begin{pmatrix} 0 & 0 & 0 & r \\ 0 & r & r & r \\ 0 & 0 & r & r \\ 0 & 0 & 0 & r \end{pmatrix}, \qquad W_{14}^* W_{13}^* W_{12}^* \hat{R} = \tilde{R} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & r & r & r \\ 0 & 0 & r & r \\ 0 & 0 & 0 & r \end{pmatrix}.$$

The matrix W is $W_{12}W_{13}W_{14}$, and these plane rotations may be accumulated in $\hat{Q}$ to form $\tilde{Q}$.

In determining the complexity of this algorithm, our goal is not to provide a detailed operation count, which would depend in a complicated way on the structure of B, but instead to show that the algorithm is $O(n^3)$. To do this note that the computation consists of at most n applications of the plane rotation algorithm described above, including the accumulation of the rotations in V and B. Moreover, the downdating of the decomposition (2.3) must be done a row at a time, of which there are at most n. Suppose that for both sources of complexity the worst case can be shown to be of order n^2, with an order constant that remains the same for the other cases. Then, since the steps must be repeated at most n times, the complexity of the entire algorithm has to be of order n^3.

The most expensive step of the plane rotation algorithm occurs at the beginning and is easily seen to be $O(n^2)$. The most expensive downdating of a row also occurs at the beginning, when only a single null vector has been found. The operation count in [18, p. 334] shows that this downdating is also of $O(n^2)$. Thus the entire algorithm is $O(n^3)$, regardless of the structure of B.

3. Error.

There are three sources of error in the above algorithm:

1. Rounding errors made in the course of the computation.
2. Initial errors in the matrix A (e.g., errors in computing or measuring A).
3. Errors in the purported null vectors.

These errors also occur in the SVD algorithm.

We can quickly dispose of rounding error. Because the algorithms in question are backward stable, the computed B is orthogonally similar to a perturbed matrix $\tilde{A}$, where the relative perturbation is of the order of the rounding unit. Thus the rounding error may be merged with the initial error. In general the latter will be the larger of the two.

The third source of error is a necessary consequence of the first two. In general a perturbed matrix will not have null vectors unless the perturbing matrix is specially structured. For example, it is well known that a generic perturbation of a zero Jordan block of order p will cause its eigenvalues to split apart and arrange themselves on the circumference of a disk of radius approximately the pth root of the size of the perturbation. Thus, a perturbation of order 10^{-16} of a zero Jordan block of order 8 will produce a cluster of eigenvalues of order 0.01. In this case, the corresponding eigenvectors will not be approximate null vectors.

Thus any null vector must be approximate and must be computed from an SVD or its equivalent. The problem then becomes one of deciding when such a vector is acceptable. The solution is to set a tolerance τ to judge the acceptability. In the SVD algorithm, any singular vector whose singular value is less than τ is deemed acceptable. In the QR/update version, any normalized vector v for which $\|Rv\|$ is less than τ is acceptable.

What this means for the overall process is that the portions of the computed B that should be zero are nonzero of size up to the order of τ. Because the computed transformations are almost exactly orthogonal, these errors can also be thrown back on A and merged with the initial errors. With this done, the computed GNSD is the exact GNSD of $A+G$, where G is the sum of the initial error, the backward rounding error, and the backward τ-error.

The problem remains of how to choose τ. Note that a choice can be unsatisfactory in one of two ways. If τ is too small, the matrix will be judged to have no null vectors; i.e., it is its own GNSD. On the other hand, if it is too large, all vectors will be deemed null, and its GNSD will be the zero matrix, with the original matrix the backward error.

Little more can be said without considering specific problems. The following parameterized problem appears to us to be flexible enough to generate interesting test cases without deviating too far from what might be expected in applications. Specifically, given input parameters k_i $(i = 1, \ldots, p)$, κ, and ρ, the matrix A has the form

$$A = XJX^{-1} + E \equiv A_{\text{true}} + E, \tag{3.1}$$

where

1. $J = \operatorname{diag}(J_{k_1}(0), J_{k_2}(0), \ldots, J_{k_p}(0))$.
2. X and X^{-1} are balanced matrices (in the sense that their elements are of roughly the same size) with condition number κ.
3. E consists of independent random normal deviates scaled so that $\|E\|/\|A_{\text{true}}\| = \rho$.

This test matrix answers to a researcher who suspects the matrix in question has a nontrivial generalized null space structure that can be reached by a balanced, possibly ill-conditioned transformation. Since, in this case, the matrix A would be computed or measured directly it is appropriate to assume that the errors would be relative to the size of A. It is, of course, possible to imagine harder problems; e.g., diagonally scaled Jordan blocks in which the elements of the first superdiagonal vary in magnitude.[5]

Returning now to the choice of τ, the norm of E is $\rho\|A_{\text{true}}\|$. Hence τ should be set somewhat larger than this quantity, say

$$\tau > \overline{\theta}\rho\|A\|, \qquad \overline{\theta} > 1.$$

The fudge factor $\overline{\theta}$ must be included to account for rounding error in determining approximate null vectors and in accumulating transformations, and it must be determined from experience.

As suggested above, τ must not be too large. It is not easy to describe a general upper limit. But since we are dealing with Jordan blocks, whose structure depends on the placement of ones on the first superdiagonal, an upper bound of at most one seems to be required. In our experiments we took τ to be

$$\tau = \sqrt{\rho\|A\|}, \tag{3.2}$$

which is the geometric mean of $\rho\|A\|$ and 1. On a logarithmic scale this choice places τ exactly in the middle of the error norm and one.

But this is not the entire story. It remains to consider how the error E reflects back to the original Jordan form. By definition, $\|E\| = \rho\|A_{\text{true}}\|$. Since $A_{\text{true}} = XJX^{-1}$ and $\|J\| = 1$, we have $\|A_{\text{true}}\| \le \|X\|\|X^{-1}\| = \kappa$. (Note that κ is just the condition number of X with respect to inversion.) Now if we set $X^{-1}AX = J + F$, then $F = X^{-1}EX$. Hence

$$\|F\| \le \kappa\|E\| \le \kappa^2\rho. \tag{3.3}$$

The upper bound $\kappa^2\rho$ is not unreasonable, since X and X^{-1} are balanced and E bears no particular relation to X. If $\kappa^2\rho$ is near one, we can expect trouble in recovering the structure of J. We will present evidence of this κ^2 effect in the next section.

4. Numerical experiments.

The purpose of this section is to compare the behavior of the QR/update and the SVD algorithms along with their limitations in recovering Jordan structures. The algorithms were coded as MATLAB functions.[6] However, null vectors in the SVD algorithm were computed by the function `svd`, and the deflations performed by matrix multiplications, for both of which MATLAB uses highly optimized BLAS and LAPACK algorithms. The QR/update algorithm, on the other hand, must apply plane rotations explicitly, a process for which MATLAB is notoriously inefficient. This precludes any meaningful timings, and the focus of this section will be on the difference in the numerical behavior of the algorithms along with their limitations.

The following is an outline of how our experiments were implemented.

1. The matrices used in the implementation were generated as described in the preceding section (see eq. (3.1)).

[5] See the interesting paper [10] by Edelman and Ma, which discusses this problem and gives an illuminating example of it.

[6] Our GNSD code is available in the supplementary materials section on the website.

```
kappa  rho          nfound  nfail  numresid     gnsdresid

1e+0   1e-5     qru   100     0   1.9893e-15   9.9130e-06
                svd   100     0   3.8282e-15   9.9130e-06

1e+1   1e-6     qru   100     0   1.0833e-15   1.7667e-06
                svd   100     0   1.9046e-15   1.7667e-06

1e+2   1e-7     qru   100     0   1.0433e-15   9.4737e-07
                svd   100     0   1.5692e-15   9.4737e-07

1e+3   1e-8     qru   100     0   9.9481e-16   6.1044e-07
                svd   100     0   1.4530e-15   6.1070e-07

1e+4   1e-9     qru    89    11   9.6409e-16   1.8697e-07
                svd    89    11   1.4582e-15   1.8695e-07
```

block sizes = 1, 2, 3, 4, 5

FIG. 1. *Comparison of QR/updated and SVD.*

2. The matrix J is specified by a sequence of block sizes k_i $(i = 1, \ldots, p)$ defining the μ_j in (1.4).
3. The random matrix X is generated in the form $X = QSP^*$, where Q and P are random orthonormal matrices and S is generated by the MATLAB expression `diag(logspace(0, log10(1/kappa),n))`.
4. The Linpack condition estimator, as described in [14, p. 296], was used to compute approximate null vectors in the QR/update algorithm (n.b., this is not the Linpack function `RCOND`).
5. The tolerance was computed by the formula (3.2).
6. Throughout, we used the spectral matrix norm, computed by the MATLAB function `norm`; i.e., $\|X\| =$ `norm(X)`.

In our experiments, we took ρ and κ as our independent variables. For given values of ρ and κ, a sample of size `sampsize` of the matrix A was processed by both the QR/update and the SVD methods. The averages for the following statistics were recorded.

1. The number of cases `nfound` in which the algorithm recovered the original structure and `nfail` the number of cases in which it failed.
2. For the successful cases, let $\hat{B} = V^*AV$ and let B be the matrix obtained from $\hat{B}$ by zeroing appropriate blocks of $\hat{B}$ to get the GNSD. Then `numresid`, the average of $\|A - V\hat{B}V^*\|/\|A\|$, is a measure of the numerical stability of the process and should be near the rounding unit, while `gnsdresid`, the average of $\|A - VBV^*\|/\|A\|$, measures the relative error in the GNSD approximation to A.
3. All of the tests were performed for a fixed block structure $k_i = i$ $(i = 1, \ldots, 5)$, for which the correct values μ_j are $6 - j$ $(j = 1, \ldots, 5)$. But other structures gave essentially the same results.

Figure 1 shows the results of five runs with sample size one hundred. The parameters κ and ρ are varied in such a way that $\kappa\rho = 10^{-5}$.

The first thing to examine in these numbers is the values of `numresid`, which determine the stability of the QR/update and the SVD algorithms. Both pass with flying colors. The relative residuals are near the rounding unit of about $2.2{\cdot}10^{-16}$.

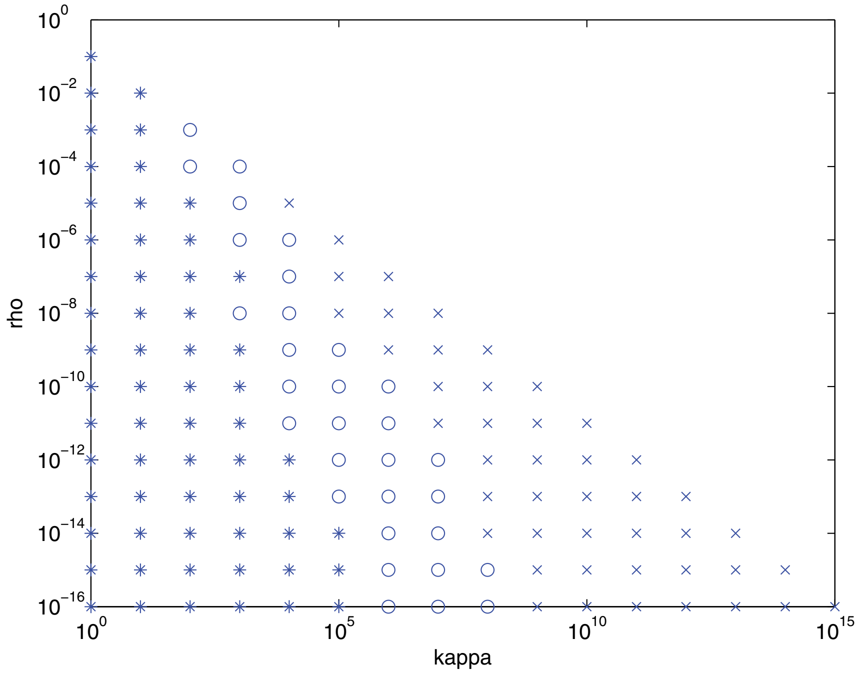

FIG. 2. *Success and failure.*

These results have been observed in other runs not reported, supporting our assertion that the QR/update algorithm is numerically stable. Naturally, so is the SVD algorithm.

The numbers `nfound` and their complements `nfail` describe the reliability of the computed GNSD. Both algorithms capture the Jordan structure as κ increases to $\kappa = 10^3$, after which both algorithms are less reliable. In another run, the product of $\rho\kappa$ was taken to be 10^{-4}. Here deterioration sets in when $\kappa = 10^3$ and grew worse for $\kappa = 10^4$. The $(\texttt{nfound}, \texttt{nfail})$ pairs were respectively $(96, 4)$ and $(63, 37)$ for both algorithms.

These numbers also support the existence of the κ^2 phenomenon described at the end of the preceding section. For the case in Figure 1 the products $\rho\kappa^2$ are 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, and 10^{-1}. For the case $\rho\kappa = 10^{-4}$, they were 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, and 10^0. In the second case some results were good even when $\rho\kappa^2 = 1$. This may be attributed to the fact that $\rho\kappa^2$ is only an upper bound for $\|F\|$ in (3.3).

The fact that the number of failures are the same for both algorithms suggests that they are computing essentially the same thing, even when they fail to capture the original Jordan structure. In several additional runs (with $\rho\kappa = 10^{-4}$) there was only a single case of the algorithms computing different structures.

To get a better handle on the κ^2 effect, consider Figure 2, which plots runs with sample size fifty for various values of κ and ρ. Runs for which the correct structure was determined for all cases are represented by the symbol `*`; those that are partially successful, by `o`; and those that failed in all cases, by `x`. The display in Figure 3 gives the common logarithm of ρ against the common logarithms of $\rho\kappa^2$ for the first `o` and

ρ	$\rho\kappa^2$		$\rho\kappa$
	o	x	x
-3	$+1$	$-$	$-$
-4	$+0$	$-$	$-$
-5	$+1$	$+3$	-1
-6	$+0$	$+4$	-1
-7	$+1$	$+3$	-2
-8	-2	$+2$	-3
-9	-1	$+3$	-3
-10	-2	$+4$	-3
-11	-3	$+3$	-4
-12	-2	$+4$	-4
-13	-3	$+3$	-5
-14	-2	$+2$	-6
-15	-3	$+3$	-6
-16	-4	$+2$	-7

FIG. 3. *Success and failure.*

the first `x` in each row. In most cases, these pairs bracket zero. On the other hand, the values for $\rho\kappa$ are all negative for `x` and their magnitudes increase as $\log_{10}\rho$ decreases; i.e., they tend to predict success where there is none, and the mispredictions become more emphatic as κ grows. All of this is consistent with the κ^2 effect.

The existence of the κ^2 effect raises a question about the choice of the tolerance τ. Should the value of τ be increased to be larger than $\rho\kappa^2$? Limited experiments suggest that this can make things worse. Perhaps the reason is that the algorithm in question is working with A not $J + F$, and raising the value of τ could mask its Jordan structure.

5. Summary. This paper began life as a derivation of some formulas relating the Drazin inverse of a matrix and the Moore–Penrose pseudoinverse. Being dissatisfied with the purely algebraic proofs of these results, we decided to see if matrix decompositions had something to say, and in the process we unwittingly rediscovered the GNSD and the SVD algorithm. The possibility of $O(n^4)$ behavior of this algorithm led us to the QR/update algorithm, which is the centerpiece of this paper.

Our numerical results confirm that the new algorithm is backward stable and produces essentially the same results as the SVD version. It should be preferable to the SVD algorithm for matrices with large zero Jordan blocks. In addition, our experiments tend to confirm that the LINPACK condition estimator is a reliable way of computing approximate null vectors of a triangular matrix.[7]

A second contribution is the introduction of a class of parameterized test problems (see eq. (3.1)) that have the flavor of real life and yet the ability to probe the limitations of GNSD algorithms as well as the stability of the GNSD itself. Regarding the latter, the discovery of the κ^2 effect was the result of experiments using this model.

[7]But perhaps less reliable for computing approximate null vectors of a general matrix from its LU factorization. See the discussion in [14, p. 297].

We should stress once more that the κ^2 effect only operates in full force when the upper bound in (3.3) is nearly attained, which may not happen in particular cases. Thus the "effect" should be regarded as indicating a potential danger, not as a death certificate.

The GNSD is well worth further study. One reason is that it is used in many algorithms that compute the Jordan form and its generalizations. Since problems with computing the much simpler GNSD will be inherited by the Jordan form, further study of the former may lead to breakthroughs in computing the latter.

Appendix A. Computing the Drazin inverse.

The purpose of this appendix is to sketch an algorithm that uses the GNSD to compute the Drazin generalized inverse of a matrix A. Let ν be the index of A. Then the Drazin inverse A^{D} is the unique matrix satisfying the following three conditions [9, 4]:

$$A^{\mathrm{D}}A^{\nu+1} = A^{\nu}, \quad A^{\mathrm{D}}AA^{\mathrm{D}} = A^{\mathrm{D}}, \quad AA^{\mathrm{D}} = A^{\mathrm{D}}A. \tag{A.1}$$

The following result provides a different characterization of A^{D} that suggests a computational algorithm.

If there is a nonsingular matrix W such that

$$W^{-1}AW = \mathrm{diag}(N, M), \tag{A.2}$$

where $N^{\nu-1} \neq 0$, $N^{\nu} = 0$, and M is nonsingular, then

$$A^{\mathrm{D}} = W\mathrm{diag}(0, M^{-1})W^{-1}.$$

The result is proved by verifying that the purported inverse satisfies the conditions (A.1).

The GNSD (1.1) provides the wherewithal to compute the Drazin generalized inverse. Let the GNSD be partitioned in the form

$$B = \begin{pmatrix} N & L \\ 0 & M \end{pmatrix},$$

where $M = B_{\nu+1,\nu+1}$ is nonsingular. By construction, $N^{\nu-1} \neq 0$ and $N^{\nu} = 0$. Only the presence of the matrix L keeps the above result from applying. However, L can be eliminated by a simple similarity transformation, as will now be shown.

Consider the similarity transformation

$$\begin{pmatrix} I & -K \\ 0 & I \end{pmatrix}\begin{pmatrix} N & L \\ 0 & M \end{pmatrix}\begin{pmatrix} I & K \\ 0 & I \end{pmatrix} = \begin{pmatrix} N & NK + L - KM \\ 0 & M \end{pmatrix}. \tag{A.3}$$

If K satisfies the Sylvester equation

$$KM - NK = L, \tag{A.4}$$

then the right-hand side of (A.3) is $\mathrm{diag}(N, M)$. But

$$\mathrm{diag}(N, M)^{\mathrm{D}} = \mathrm{diag}(0, M^{-1}).$$

If the similarity transformation in (A.3) is inverted, the result is

$$\begin{pmatrix} N & L \\ 0 & M \end{pmatrix}^{\mathrm{D}} = \begin{pmatrix} 0 & KM^{-1} \\ 0 & M^{-1} \end{pmatrix}.$$

Now a necessary and sufficient condition for the existence of a unique solution of the Sylvester equation (A.4) is that the eigenvalues of N and M be disjoint. But N is nilpotent and hence has only zero eigenvalues, while M is nonsingular and has only nonzero eigenvalues. Hence (A.4) has a unique solution. Moreover, since N is triangular the equation may be solved by a variant of the Bartels–Stewart method [2] that involves only solutions of linear systems whose matrix is M. But if the GNSD is computed by the QR/update method, a QR factorization of M will be at hand to solve the systems.

We note that (A.2) leads to a delightfully simple formula relating the Drazin inverse A^{D} to any generalized inverse A^- satisfying $AA^-A = A$. Partition $W = (W_1\ W_2)$ and $(W^{-1})^* = (Z_1\ Z_2)$ conformally with the matrix $\mathrm{diag}(N, M)$, so that $Z_1^*W_1 = I$, $Z_2^*W_2 = I$, $Z_1^*W_2 = 0$, and $Z_2^*W_1 = 0$. It follows that $A = W_1NZ_1^* + W_2MZ_2^*$ and $A^{\mathrm{D}} = W_2M^{-1}Z_2^*$. Now define P as the projection $W_2Z_2^*$, and notice that $AA^{\mathrm{D}} = P = A^{\mathrm{D}}A$. Therefore,

$$PA^-P = (A^{\mathrm{D}}A)A^-(AA^{\mathrm{D}}) = A^{\mathrm{D}}(AA^-A)A^{\mathrm{D}} = A^{\mathrm{D}}AA^{\mathrm{D}} = A^{\mathrm{D}}.$$

While the formula $A^{\mathrm{D}} = PA^-P$ is not new (see [13, eq. 12]), we were not able to find it in any of the books on generalized inverses that we consulted, so it seems safe to say it is not as well known as it should be.

Appendix B. Two applications. Here we briefly describe two applications where we have found the computation of the GNSD to be useful.

B.1. An application arising in surface subdivision. The design of smooth surfaces using subdivision algorithms, a common technique used in computer graphics, leads to certain eigenvalue optimization problems; see Grundel's Ph.D. thesis [12] for details. Computations described there in chapters 2 and 4 for a triangular mesh and a quadrilateral mesh, respectively, led to two interesting matrices. The first, which was discovered by a combination of numerical and analytical techniques, is given by

$$\begin{pmatrix}
z & s & s & s & t & u & t & u & t & u \\
a & b & c & c & e & d & 0 & 0 & 0 & d \\
a & c & b & c & 0 & d & e & d & 0 & 0 \\
a & c & c & b & 0 & 0 & 0 & d & e & d \\
3/32 & 7/16 & 3/32 & 3/32 & 3/32 & 3/32 & 0 & 0 & 0 & 3/32 \\
9/64 & 39/128 & 39/128 & 3/64 & 3/128 & 9/64 & 3/128 & 1/128 & 0 & 1/128 \\
3/32 & 3/32 & 7/16 & 3/32 & 0 & 3/32 & 3/32 & 3/32 & 0 & 0 \\
9/64 & 3/64 & 39/128 & 39/128 & 0 & 1/128 & 3/128 & 9/64 & 3/128 & 1/128 \\
3/32 & 3/32 & 3/32 & 7/16 & 0 & 0 & 0 & 3/32 & 3/32 & 3/32 \\
9/64 & 39/128 & 3/64 & 39/128 & 3/128 & 1/128 & 0 & 1/128 & 3/128 & 9/64
\end{pmatrix},$$

where $a = 233/896$, $b = 248/896$, $c = 171/896$, $d = 29/896$, $e = 15/896$, $z = 69/448$, $s = 2101/9632$, $t = 295/19264$, and $u = 1403/28896$. This matrix is singular and, because it is known exactly, its Jordan form can be computed using standard techniques, showing that the zero eigenvalue has multiplicity 4, with one Jordan block of order 2 and two of order 1. The left panel of Figure 4 shows the index computed by the GNSD code described in section 4 as a function of the tolerance τ. GNSD correctly determines that the index ν is 2 with $\mu_1 = 3$ and $\mu_2 = 1$ for all values of τ except those that are so large that they swamp the data or so small that rounding errors dominate the computation.

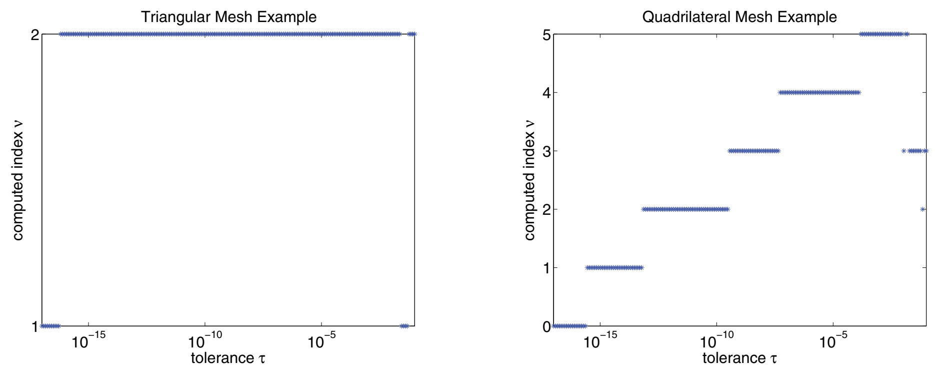

FIG. 4. *Computed index for triangular mesh and quadrilateral mesh matrices as a function of the tolerance τ.*

In contrast, the second matrix, which has order 18, was computed by numerical optimization and is not known exactly.[8] The right panel of Figure 4 shows the computed index for this matrix. The largest computed index ν is obtained for τ approximately in the range 10^{-4} to 10^{-2}, with $\nu = 5$ and $\mu_1 = 4$, $\mu_2 = 4$, $\mu_3 = 2$, $\mu_4 = 1$, $\mu_5 = 1$, indicating that the approximate zero eigenvalue has Jordan blocks of size 5, 3, 2, and 2, as had been conjectured using a heuristic argument [12, p. 66].

B.2. An application arising in an investigation of Crouzeix's conjecture. Crouzeix's conjecture [7] is that $\|p(A)\|_2 \leq 2\|p\|_{W(A)}$ for any polynomial p and any square matrix A, where the norm on the right-hand side of the inequality is the maximum of $|p(z)|$ over all $z \in W(A)$, the field of values of A. In [6, p. 3254], Choi showed that the inequality is tight when $p(z) = z^n$ and A is an $(n+1) \times (n+1)$ matrix with just one nonzero diagonal, namely, $(\sqrt{2}, 1, \ldots, 1, \sqrt{2})$ on the first superdiagonal. Recently, numerical experiments were conducted,[9] searching for minimizers of the "Crouzeix ratio" $\|p\|_{W(A)}/\|p(A)\|_2$; if Crouzeix's conjecture is true, the globally minimal value is 0.5. In one experiment of particular interest, p was fixed by $p(z) = z^n$, while A was set to a variable $(n+1) \times (n+1)$ upper Hessenberg matrix. Although the ratio 0.5 was often obtained, at first the computed minimizers offered little insight. However, application of the GNSD code to these matrices unexpectedly revealed that their upper triangular forms on the left-hand side of (1.1) are nearly in Choi's superdiagonal form, suggesting a possible converse to Choi's result that might yield insight into Crouzeix's conjecture.

Appendix C. Characterizing the generalized null spaces. Here, for completeness, we prove the characterization (1.3) of the spaces $\mathcal{V}_j$, a result that is implicit in [11], for example. It may be assumed that the matrix A is already in the GNSD form (1.1), in which case the matrices V_j are simply the columns of the identity matrix corresponding to the jth block column of B. The last block column of B may also be ignored, since for any k the matrix $B^{(k)}_{\nu+1,\nu+1}$ is nonsingular, and the corresponding block column cannot contribute to the null spaces of the powers of B. Thus it is

[8]See supplementary materials on the website for the MATLAB data file defining its best known approximation.

[9]In work in progress by the second author in collaboration with A. Greenbaum et al.

sufficient to consider the matrix

$$C = \begin{pmatrix} 0 & C_{12} & C_{13} & \cdots & C_{1,\nu-1} & C_{1,\nu} \\ 0 & 0 & C_{23} & \cdots & C_{2,\nu-1} & C_{2,\nu} \\ 0 & 0 & 0 & \cdots & C_{3,\nu-1} & C_{3,\nu} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & C_{\nu-1,\nu} \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}. \tag{C.1}$$

Note that the form of C already implies that $\mathcal{V}_1$ satisfies the characterization (1.3) because the matrices on the first superdiagonal are of full rank and hence C can have no null vectors other than those in $\mathcal{V}_1$. The question, then, is what happens to the powers of C. The following lemma shows that $\mathcal{V}_1 \cup \mathcal{V}_2$ contains all the null vectors of C^2 and similarly for the higher powers of C.

LEMMA 1. *Let C have the form* (C.1), *where the diagonal blocks are square, and the blocks on the first superdiagonal are of full column rank. Then all the blocks of C^j to the southwest of the jth superdiagonal are zero, and the blocks on the jth superdiagonal are of full column rank.* A formal proof of this lemma based on the formulas for matrix multiplication is tedious and obscures what is going on. Instead, the truth of the lemma will be shown for C^2 and C^3 when $\nu = 6$, after which the general validity of the lemma should be evident.

By direct computation, it follows that

$$C^2 = \begin{pmatrix} 0 & 0 & C_{13}^{(2)} & X & X & X \\ 0 & 0 & 0 & C_{24}^{(2)} & X & X \\ 0 & 0 & 0 & 0 & C_{35}^{(2)} & X \\ 0 & 0 & 0 & 0 & 0 & C_{46}^{(2)} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where $C_{i,i+2}^{(2)} = C_{i,i+1}C_{i+1,i+2}$. The diagonal blocks of the product remain square, while the elements on the second superdiagonal are pairwise products of the superdiagonal elements of C. Since the products of matrices of full column rank are themselves of full column rank, the $C_{i,i+2}$ must likewise be of full column rank.

The next powering illustrates the induction step of a formal proof. Again by direct computation it follows that

$$C^3 = CC^2 = \begin{pmatrix} 0 & 0 & 0 & C_{14}^{(3)} & X & X \\ 0 & 0 & 0 & 0 & C_{25}^{(3)} & X \\ 0 & 0 & 0 & 0 & 0 & C_{36}^{(3)} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where $C_{i,i+3}^{(3)} = C_{i,i+1}C_{i+1,i+3}^{(2)}$. Again, the diagonal blocks of C^3 are square, and—here is the induction step—the matrices $C_{i,i+1}$ and $C_{i+1,i+3}^{(2)}$ are of full column rank. Hence their products, the $C_{i,i+3}^{(3)}$, are of full column rank.[10]

[10]It is perhaps worth noting that if C^2 had been *postmultiplied* by C, the result would give different formulas for the $C_{i,i+3}^{(3)}$. Equating the different forms of the blocks gives nontrivial algebraic identities; e.g., $C_{13}^{(2)}C_{34} = C_{12}C_{24}^{(2)}$.

REFERENCES

[1] K. M. Anstreicher and U. G. Rothblum, *Using Gauss–Jordan elimination to compute the index, generalized nullspaces, and Drazin inverse*, Linear Algebra Appl., 85 (1987), pp. 221–239.

[2] R. H. Bartels and G. W. Stewart, *Algorithm* 432*: The solution of the matrix equation* $AX + XB = C$, Comm. ACM, 8 (1972), pp. 820–826.

[3] Th. Beelen and P. Van Dooren, *Computational aspects of the Jordan canonical form*, in Reliable Numerical Computation, M. G. Cox and S. Hammarling, eds., Oxford University Press, New York, 1990, pp. 57–72.

[4] S. L. Campbell and C. D. Meyer, Jr., *Continuity properties of the Drazin pseudoinverse*, Linear Algebra and Appl., 10 (1975), pp. 77–83.

[5] T. F. Chan, *Rank revealing QR factorizations*, Linear Algebra Appl., 88/89 (1987), pp. 67–82.

[6] D. Choi, *A proof of Crouzeix's conjecture for a class of matrices*, Linear Algebra Appl., 438 (2013), pp. 3247–3257.

[7] M. Crouzeix, *Numerical range and functional calculus in Hilbert space*, J. Funct. Anal., 244 (2007), pp. 668–690.

[8] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, *Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization*, Math. Comp., 30 (1976), pp. 772–795.

[9] M. P. Drazin, *Pseudo-inverses in associative rings and semigroups*, Amer. Math. Monthly, 65 (1958), pp. 506–514.

[10] A. Edelman and Y. Ma, *Staircase failures explained by orthogonal versal forms*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1004–1025.

[11] G. H. Golub and J. H. Wilkinson, *Ill-conditioned eigensystems and the computation of the Jordan canonical form*, SIAM Rev., 18 (1976), pp. 578–619.

[12] S. Grundel, *Eigenvalue Optimization in* C^2 *Subdivision and Boundary Subdivision*, Ph.D. thesis, New York University, New York, NY, 2011; available online at http://cs.nyu.edu/overton/phdtheses/sara.pdf.

[13] R. E. Hartwig, *More on the Souriau–Frame algorithm and the Drazin inverse*, SIAM J. Appl. Math., 31 (1976), pp. 42–46.

[14] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd. ed., SIAM, Philadelphia, 2002.

[15] B. Kågström and A. Ruhe, *An algorithm for numerical computation of the Jordan normal form of a complex matrix*, ACM Trans. Math. Software, 6 (1980), pp. 398–419.

[16] V. N. Kublanovskaja, *A method for solving the complete problem of eigenvalues of a degenerate matrix*, USSR Comput. Math. Math. Phys., 4 (1968), pp. 1–16. Originally appeared in Ž. Vyčisl. Math. Math. Fiz, 6 (1966), pp. 611–620.

[17] A. Ruhe, *An algorithm for numerical determination of the structure of a general matrix*, BIT, 10 (1970), pp. 196–216.

[18] G. W. Stewart, *Matrix Algorithms* I: *Basic Decompositions*, SIAM, Philadelphia, 1998.

[19] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

[20] P. Van Dooren, *The computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 27 (1979), pp. 103–140.

[21] K. Yoo and H. Park, *Accurate downdating of a modified Gram-Schmidt QR decomposition*, BIT, 36 (1996), pp. 166–181.