ON THE EXISTENCE AND COMPUTATION OF LU-FACTORIZATIONS WITH SMALL PIVOTS

Tony F. Chan

Technical Report # 227
March 22, 1982

Computer Science Department, Yale University, Yale Station, New Haven, CT 06520.
This work was supported by the Department of Energy under grant DE-AC02-81ER10996.

Table of Contents

Abstract

Let A be a n by n matrix which may be singular with a one-dimensional null space, and consider the LU-factorization of A. When A is exactly singular, we show conditions under which a pivoting strategy will produce a zero n-th pivot. When A is not singular, we show conditions under which a pivoting strategy will produce a n-th pivot that is $O(\sigma_n)$ or $O(\kappa^{-1}(A))$, where σ_n is the smallest singular value of A and $\kappa(A)$ is the condition number of A. These conditions are expressed in terms of the elements of A^{-1} in general but reduce to conditions on the elements of the singular vectors corresponding to σ_n when A is nearly or exactly singular. They can be used to build a 2-pass factorization algorithm which is *guaranteed* to produce a small n-th pivot for nearly singular matrices. As an example, we exhibit a LU-factorization of the n by n upper triangular matrix

$$T = \begin{vmatrix} 1 & -1 & -1 & . & . & -1 \\ & 1 & . & & & . \\ & & 1 & . & & . \\ & & & . & . & . \\ & 0 & & & . & . \\ & & & & . & -1 \\ & & & & & 1 \end{vmatrix}$$

that has a n-th pivot equal to $2^{-(n-2)}$.

1. Introduction

The LU-factorization PAQ = LU (in this paper, P and Q always denote permutation matrices, L is always *unit* lower triangular and U upper triangular) of a general m by n matrix A plays an important role in computational linear algebra. It always exists and can be found efficiently by Gaussian Elimination or its variants. On the other hand, there may be many LU-factorizations for a given matrix. However, the LU factorization is *unique* once the permutations P and Q are fixed. In this paper, we shall be concerned only with *square* n by n matrices, with primary interest on the *nearly singular* case.[1] When A is exactly singular, then *any* LU factorization of A has a zero on the diagonal of U (i.e. a zero pivot). However, when A is nearly singular, it is well known that U may

[1]We shall assume that the nullity of A is one in this paper.

not have any small diagonal elements. Note that the diagonal elements of U are the eigenvalues of U. Golub and Wilkinson [10] showed that the smallest eigenvalue λ_n of any matrix can be bounded in terms of the singular values σ_i as follows:

$$|\lambda_n| \leq \sigma_1 (\sigma_n/\sigma_1)^{1/n},$$

and this bound is the *best* possible in general. Thus, although (σ_n/σ_1) may be small, λ_n may not be. In fact, a well-known example is the matrix T defined earlier. For large n, T is nearly singular and both the partial and complete pivoting strategies will produce T as the U matrix in the LU-factorization of T. Obviously, there are no small elements on the diagonal of T.

In this paper, we show that, for any square matrix A, there always exists a LU factorization with a 'small' element in the last position on the diagonal of U. Here 'small' means $u_{n,n} = O(\kappa^{-1}(A))$ or $O(\sigma_n)$, where $\kappa(A)$ is the condition number of A in some norm, and σ_n is the smallest singular value of A.[2] *Thus, we can always find a LU-factorization with a* $u_{n,n}$ *that is as small as A is nearly singular.* For a given matrix, there may be many such LU-factorizations. We show conditions on A which show how many of these factorizations are possible. These conditions are expressed in terms of the elements of A^{-1} in general and reduce to conditions on the elements of the singular vectors corresponding to σ_n when A is nearly or exactly singular. These conditions also show that matrices which are nearly singular but which the commonly used pivoting strategies do not produce a small $u_{n,n}$ all have a very special pattern to their inverses and their smallest singular vectors. Moreover, simple permutations of these matrices will produce small pivots with the usual pivoting strategies. Therefore, they are in some sense rare and relatively harmless. Based on these conditions, we propose a 2-pass algorithm which is *guaranteed* to produce a LU-factorization of *any* given matrix with a n-th pivot that is as small as A is singular. The extra work involved is usually just a few more backsolves and at worst one more factorization. A related theoretical question is how to permute the rows and columns of a *singular* matrix A so as to obtain LU-factorizations with $u_{n,n} = 0$. We show that these permutations can be expressed in terms of the positions of the nonzero elements of the smallest singular vectors of A and are consistent with the conditions for producing small pivots of nearly singular matrices as these matrices tend to be exactly singular.

---

[2]We shall use upper case letters for denoting matrices and the corresponding lower case with subscripts for denoting elements of matrices.

The existence of a small pivot reveals a great deal about the null space of A. Such a factorization can be used to determine the rank of A and for determining the approximate left and right null vectors of A without inverse iterations [12, 13]. It can be used to compute the pseudo-inverse of A [16], to solve least squares problems [5, 17, 16] and to solve underdetermined linear systems [6]. Another important application is to computing *deflated solutions* and *deflated decompositions* of solutions of nearly singular linear systems [3, 13, 19] which arise in numerical continuaton methods for solving nonlinear systems [2, 4, 13, 18, 20]. Many numerical methods have been proposed which are designed to exploit such LU-factorizations. Naturally, all of these methods depend on the ability of some procedures for producing such factorizations. Therefore, it is important to better understand both the theoretical questions of existence and the practical questions of computing such factorizations.

In Section 2, we motivate and outline the basic strategy used to find permutations P and Q such that there *exist* LU-factorizations for PAQ. In Section 3, we treat the exactly singular case and in Section 4, we treat the nonsingular case. We present the 2-pass algorithm in Section 5. In Section 6, we use the conditions developed in Section 4 to explicitly construct for the matrix T a LU-factorization exhibiting a $u_{n,n} = 2^{-(n-2)}$. In Section 7, we show actual numerical computations with the matrix T and another example of Wilkinson's [21]. We conclude with a few remarks on possible extensions of this work in Section 8.

2. Existence

The basic observation which allows the permutations P and Q to be computed is the result of the following lemma.

Lemma 1: (a) If A is *nonsingular* and PAQ has a LU-factorization of the form

$$PAQ = \begin{vmatrix} L_1 & 0 \\ v^T & 1 \end{vmatrix} \begin{vmatrix} U_1 & w \\ 0 & \epsilon \end{vmatrix}, \quad (1)$$

then we can perturb the (n,n)-th element of PAQ by ϵ to make it *singular*.

(b) If A is *singular* and PAQ has a LU-factorization of the form (1) with $\epsilon = 0$, then we can perturb the (n,n)-th element of PAQ to make A *nonsingular*.

Proof: Multiplying the factors in (1) will reveal that ϵ only enters into the expression for the (n,n)-th element of PAQ and therefore changing ϵ (to zero in (a) and to nonzero in (b)) will only affect the (n,n)-th element of PAQ.

Our strategy is based on the converse of Lemma 1, i.e. we want to find elements of A which can be perturbed alone to change the singularity (rank) of A. For A nonsingular, we want to find elements which we can perturb by the *smallest* amount possible. We therefore make the following definition.

Definition 2: Let $C_1 = \{a_{i,j} \mid \text{Rank}(A) \text{ can be changed by perturbing } a_{i,j} \text{ alone}\}$.

Once these elements are found, we can then use permutation matrices P and Q to move them to the (n,n)-th position in PAQ. Finally, we have to construct the desired LU-factorization of the permuted matrix. For this last step, we need the following lemma.

Lemma 3: Let A be represented in the partitioned form:

$$A = \begin{vmatrix} S & q \\ & \\ p^T & d \end{vmatrix}, \tag{2}$$

where S is (n-1) by (n-1) and p and q are vectors. Then we can change the singularity of A by perturbing the element d *if and only if* S is nonsingular.

Proof: By the cofactor expansion, the determinant of A, denoted by det(A), is equal to d times det(S) plus terms independent of d. The fact that we can change the rank of A by changing d means that det(S) has to be nonzero. On the other hand, if det(S) $\neq$ 0, then we can change d to make the product d det(S) cancel the rest of the terms to make A singular.

Now we can state our basic theorem on the existence of a LU-factorization for PAQ.

Theorem 4: PAQ has a LU-factorization *if and only if* P and Q permute an element of C_1 to the (n,n)-th position of PAQ.

Proof: The only if part is exactly Lemma 1. We shall now prove the if part. Write PAQ in the form of (2). Then by Lemma 3, S is nonsingular. Therefore, S has a LU-factorization $P_S S Q_S = L_S U_S$, where U_S has nonzero diagonal elements. It can then be easily verified that PAQ has the following LU-factorization:

$$PAQ = \begin{vmatrix} L_S & 0 \\ p^T U_S^{-1} & 1 \end{vmatrix} \begin{vmatrix} U_S & L_S^{-1} q \\ 0 & u_{n,n} \end{vmatrix}, \quad (3)$$

where $u_{n,n} = d - p^T U_S^{-1} L_S^{-1} q$ and the permutations P_S and Q_S are assumed to have been applied to p and q and absorbed into P and Q.

Note that if A is exactly singular, then $u_{n,n}$ must be zero.

3. The Singular Case

Let A be a *singular* matrix with a one dimensional null space. In this section, we show how to find permutations P and Q such that PAQ has a LU-factorization with $u_{n,n} = 0$. Note that, by Theorem 4, we only have to find the elements of C_1.

Definition 5: Let the *Singular Value Decomposition* (SVD) of A be $A = X \Sigma Y^T$, where X and Y are *unitary* matrices, and let the columns of X be $\{x_1,...,x_n\}$ and the columns of Y be $\{y_1,...,y_n\}$ and $\Sigma = \text{Diagonal}\{\sigma_1,...,\sigma_n\}$.

Note that if A is singular, then $\sigma_n = 0$ and x_n and y_n are the ***left and right null vectors*** of A respectively.

We need a preliminary lemma.

Lemma 6: Let $D = \text{Diagonal}\{d_1, \ldots, d_{n-1}, 0\}$, and v and w be arbitrary vectors. Then the following identities hold.

(a) $\det(I + vw^T) = 1 + w^T v$,

(b) $\det(D + vw^T) = (\prod_{i=1}^{n-1} d_i)\, v_n w_n$.

Proof: See Appendix.

The next lemma shows that C_1 is related to the nonzero elements of the singular vectors corresponding to σ_n.

Definition 7: Define $C_2 = \{a_{i,j} \mid (x_n)_i (y_n)_j \neq 0\}$.[3]

[3]We use the notation $(v)_k$ to denote the k-th element of the vector v.

Lemma 8: If A is singular and has a one dimensional null space, then $C_1 \equiv C_2$ and is non-empty.

Proof: Consider perturbing the (i,j)-th element of A by δ. We have

$$\det(A + \delta e_i e_j^T)$$

$$= \det(X) \det(\Sigma + \delta (X^T e_i) (Y^T e_j)^T) \det(Y)$$

$$= \det(X) \left(\prod_{i=1}^{n-1} \sigma_i \right) \delta (x_n)_i (y_n)_j \det(Y), \text{ by Lemma 6, part (b)}$$

and thus $\det(A + \delta e_i e_j^T) \neq 0$ if and only if $(x_n)_i (y_n)_j \neq 0$ since det(X) and det(Y) are nonzero. Since x_n and y_n have Euclidean norms equal to one, they are non-trivial and thus C_2 is non-empty.

We thus arrive at the main result of this section.

Theorem 9: If A has a one dimensional null space, then PAQ has a LU-factorization with $u_{n,n} = 0$ ***if and only if*** P and Q permute an element in C_2 to the (n,n)-th position in PAQ. Moreover, there always exists at least one such factorization for any A with a one dimensional null space.

The set C_2 can be viewed as a *coloring* of the elements of A and a given pivoting strategy can be viewed as applying permutations on this coloring. It is well-known that the complete pivoting strategy (CP) will always produce a LU-factorization with $u_{n,n} = 0$ but partial pivoting (PP) may not. The fact that CP will always work is consistent with (but not a result of) Theorem 9. The following Theorem states the conditions under which PP will *not* work.

Theorem 10: The partial pivoting (PP) strategy will produce a LU-factorization with $u_{n,n} = 0$ ***only if*** C_2 contains at least one element from the *last column* of A.

4. The Nonsingular Case

Assume that A is *nonsingular*. In this section, we show how to find permutations P and Q such that PAQ has a LU-factorization with a $u_{n,n}$ that is as small as A is singular. First, we show that in this case C_1 is related to the nonzero elements of A^{-1}.

Definition 11: Let $M \equiv A^{-1}$. Define $C_3 \equiv \{a_{i,j} \mid m_{j,i} \neq 0\}$.

Lemma 12: If A is nonsingular, then $C_1 \equiv C_3$. Moreover, if $a_{i,j} \in C_3$, then

$\det(A - m_{j,i}^{-1} e_i e_j^T) = 0.$

Proof: Consider perturbing the (i,j)-th element of A by δ. Thus $\det(A - \delta e_i e_j^T) = \det(A^{-1})$ $\det(I - \delta A^{-1} e_i e_j^T) = \det(A^{-1})\,(1 - \delta m_{j,i})$ by Lemma 6, from which the results follow easily.

Next, we prove a result that relates the size of $u_{n,n}$ to the size of perturbations needed to change the rank of A.

Lemma 13: If $\det(A - \epsilon e_i e_j^T) = 0$ with $\epsilon \neq 0$, and P and Q permute the (i,j)-th element of A to the (n,n)-th element of PAQ, then PAQ has a LU-factorization with $u_{n,n} = \epsilon$.

Proof: Write PAQ in a partitioned form similar to (2), and $\widetilde{A} \equiv P(A - \epsilon e_i e_j^T)Q$ in a similar form except that d is replaced by $d - \epsilon$. By Theorem 4, $\widetilde{A}$ has a LU-factorization similar to (3) with $\widetilde{u}_{n,n} = d - \epsilon + p^T S^{-1} q = 0$. On the other hand, by a construction similar to that used in the proof of Theorem 4, it can be shown that PAQ has a LU-factorization with $u_{n,n} = d + p^T S^{-1} q$ and therefore $u_{n,n} = \epsilon$.

Combining the last two lemmas, we have the following result.

Theorem 14: If A is ***nonsingular***, then PAQ has a LU-factorization ***if and only if*** P and Q permute an element $a_{i,j} \in C_3$ to the (n,n)-th position of PAQ. Moreover, the resulting LU-factorization has $u_{n,n} = m_{j,i}^{-1}$.

To produce a small pivot, we have to look for the large elements of A^{-1}.

Definition 15: Define $||A||_L = \max_{i,j} |a_{i,j}|,\ 1 \leq i, j \leq n.$

It can easily be verified that $||.||_L$ is a matrix norm, and satisfy the following norm-equivalence.

Lemma 16:

(a) $(1/n)\,||A||_\infty \leq ||A||_L \leq ||A||_\infty.$ (4)

(b) $||A||_\infty \kappa_\infty^{-1}(A) \leq ||A^{-1}||_L^{-1} \leq n||A||_\infty \kappa_\infty^{-1}(A),$ (5)

where $\kappa_\infty^{-1}(A) \equiv ||A||_\infty ||A^{-1}||_\infty.$

Proof: Straight-forward.

We are primarily interested in the upper bound in (b) in the above lemma. It shows that the largest element of A^{-1} in absolute value is $O(\kappa_\infty(A))$.

The next definition defines the set of large elements of A^{-1}.

Definition 17: Let $r \geq 1$ be a real positive scalar. Define $C_4(r) \equiv \{a_{i,j} \mid m_{j,i} \neq 0$ and $m_{j,i}^{-1} \leq rn||A||_\infty \kappa_\infty^{-1}(A)\}$.

Lemma 18: The size of $C_4(r)$ is a ***non-decreasing*** function of r and $C_4(1)$ is ***nonempty.***

Proof: Follows directly from Lemma 16.

We can characterize the LU-factorizations of A with small $u_{n,n}$ by the *coloring* C_4.

Theorem 19: If A is ***nonsingular,*** then PAQ has a LU-factorization with $|u_{n,n}| \leq rn||A||_\infty \kappa_\infty^{-1}(A)$ ***if and only if*** P and Q permute an element of $C_4(r)$ to the (n,n)-th position of PAQ.

Proof: The *if* part follows from the *if* part of Theorem 14 and the fact that $C_4(r)$ is a subset of C_3. The ***only if*** part follows from the rest of Theorem 14 and the definition of $C_4(r)$.

We can also characterize the LU-factorizations of A with small $u_{n,n}$ by the the singular vectors x_n and y_n corresponding to σ_n.

Definition 20: Let $r \geq 1$ be a real positive scalar. Define $C_5(r) \equiv \{a_{i,j} \mid |(x_n)_i(y_n)_j| \geq 1/rn\}$, and $A^+ \equiv \sum_{i=1}^{n-1} \sigma_i^{-1} y_i x_i^T$.

Note that A^+ is the ***pseudo-inverse*** of A with σ_n set to zero.

Lemma 21: The size of $C_5(r)$ is a ***non-decreasing*** function of r and $C_5(1)$ is ***nonempty.***

Proof: Since x_n and y_n have Euclidean norm equal to one, they satisfy $||x_n||_\infty \geq 1/\sqrt{n}$ and $||y_n||_\infty \geq 1/\sqrt{n}$, and thus there is at least one element in $C_5(1)$.

Theorem 22: If P and Q permute an element of $C_5(r)$ to the (n,n)-th position of PAQ, then PAQ has a LU-factorization with $u_{n,n}$ satisfying the following bounds:

(a) $|u_{n,n}| \leq \sigma_n rn(1 - rn^2(\sigma_n/\sigma_{n-1}))^{-1}$, (6)

(b) $|u_{n,n}| \leq \sigma_n rn(1 - rn\sigma_n||A^+||_L)^{-1}$, (7)

provided the quantities inside the brackets are positive.

Proof: It can be shown that if the quantities inside the brackets are positive, then $C_5(r)$ is a subset of C_3 and therefore the LU-factorizations exist by Theorem 14. The bounds are obtained by finding lower bounds for the absolute values of the elements of A^{-1}

corresponding to elements in $C_5(r)$.

In the limit as $\sigma_n \to 0$, $C_5(r) \to C_2$ for large enough r and the bounds in Theorem 22 show that $u_{n,n} \to 0$. Thus, the results of Theorem 22 reduce to that of the *if* part of Theorem 9.

Theorem 19 is more general but requires knowledge about A^{-1}. Theorem 22 is more useful when A is nearly singular because it is more likely that it will be applicable and because it only uses the singular vectors x_n and y_n. Unfortunately, the bounds are not tight in general. However, $C_5(r)$ can be used to *indicate* where the large elements of A^{-1} are located, since $A^{-1} = A^{+} + \sigma_n^{-1} y_n x_n^T$, the last term with σ_n^{-1} will tend to dominate the first as $\sigma_n \to 0$.

Most pivoting strategies in use are designed to control numerical stability and/or sparseness structures rather than to produce a small (or zero) pivot at the (n,n)-th position of U. Given a pivoting strategy, the chance that it will produce a small $u_{n,n}$ for a given matrix seems to depend on the size of the sets $C_4(r)$ and $C_5(r)$. Without any *a priori* knowledge about either the matrix or the pivoting strategy, the chance that a pivotin strategy will choose an element from $C_4(r)$ or $C_5(r)$ increases with the size of these sets. Conversely, if these sets contain only a few elements for relatively small values of r, then it is highly likely that a pivoting strategy will not produce a small $u_{n,n}$. The following theorem states the conditions under which PP will *not* produce a small $u_{n,n}$.

Theorem 23: The partial pivoting (PP) strategy will produce a LU-factorization with $|u_{n,n}| \leq rn||A||_\infty \kappa_\infty^{-1}(A)$ *only if* $C_4(r)$ contains at least one element from the *last column* of A.

For nearly singular matrices, the size of $C_5(r)$ depends on the number of elements with large absolute values in the approximate null vectors. Many of the known examples of nearly singular matrices for which the common pivoting strategies fail to produce small pivots have very *sparse* colorings corresponding to $C_4(r)$ and $C_5(r)$. These matrices are *rare* in the sense that their inverses and null vectors have very *skew* distributions of the size of their elements, namely, only a few elements of A^{-1} are $O(\kappa(A))$ and only a few elements of x_n and y_n are $O(1)$. Fortunately, they are relatively *harmless* since the common pivoting strategies have no problems with simple permutations of them. We shall see some examples in the next few sections.

5. Algorithms

Based on the results of the last two sections, we can build an efficient 2-pass algorithm for computing a LU-factorization of a given matrix A with a small $u_{n,n}$. The algorithm takes the following general form:

Algorithm SP:

1. Compute the LU-factorization of A by some conventional pivoting strategy (e.g. PP).
2. Estimate the condition number of A [7, 8, 15].
3. If $|u_{n,n}|$ is $O(||A||\kappa^{-1}(A))$ then *done*.
4. Use a few steps of inverse iteration to find approximate singular vectors x_n and y_n.
5. Determine the set $C_5(r)$ for some reasonable value of r.
6. Repeat the following until *found* or $C_5(r)$ is *empty* :
   a. Find the largest element (in absolute value), say $a_{k,l}$, remaining in $C_5(r)$.
   b. Compute $m_{l,k}$ by solving for the k-th column z of A^{-1} from $Az = e_k$, and extracting the l-th component of z.
   c. If $m_{l,k} = O(||A||^{-1}\kappa(A))$ then set *found* to *true* else discard $a_{k,l}$ from $C_5(r)$.
7. If *not found* then compute A^{-1} and find the largest element $m_{l,k}$.
8. Find P and Q that will permute $a_{k,l}$ to the (n,n)-th position of PAQ.
9. Compute the LU-factorization of PAQ and *force* the pivoting strategy into *not* moving the (n,n)-th element of PAQ.

Algorithm SP will usually succeed at Step 3, unless A is nearly singular and A^{-1} has a very skew distribution of the sizes of its elements. If the pivoting strategy fails to produce a small $u_{n,n}$ when A is nearly singular, then the set $C_5(r)$ will *most likely* contain an element with a large $m_{l,k}$, so that the inner loop at Step 6 will converge after one or two iterations. If this step fails, then we have to compute A^{-1} which is more expensive but is *guaranteed* to work by Theorem 19.

If we do not have to resort to computing A^{-1}, then Algorithm SP will cost at worst two factorizations and a few backsolves at Steps 2, 4 and 6.b. For a general *dense* n by n matrix, a LU-factorization costs $n^3/3$ floating point operations, a backsolve costs n^2 operations and computing the inverse costs n^3 operations. On the other hand, a full SVD costs about $10n^3$ operations [1, 9]. The storage overhead of Algorithm SP is an extra copy of the original A and a few vectors. Thus, in situations where determining the rank is important, but where a full SVD is not needed, Algorithm SP may be competitive.

For problems where the same LU-factorization may be used many times (e.g. many right hand sides), the extra cost may not be significant. Moreover, in situations where a *sequence* of related A's have to be factored (e.g. in numerical continuation methods around singular points [2, 3, 11, 14, 18]), Algorithm SP has to be executed only *once*, as the permutations P and Q produced by it can be re-used by the nearby problems. If A has special structures (e.g. banded), then the permutations P and Q should be determined to preserve as much of the structures as possible.

6. An Example

In this section, we shall demonstrate the effectiveness of Algorithm SP by applying it *algebraically* to the matrix T defined earlier to produce a LU-factorization with a $u_{n,n} = 2^{-(n-2)}$.

Since T is upper triangular, it is easy to find its inverse:

$$T^{-1} = \begin{vmatrix} 1 & 1 & 2 & 4 & . & . & 2^{n-2} \\ & 1 & 1 & 2 & . & . & . \\ & & 1 & . & . & . & . \\ & & & . & 1 & 2 & 4 \\ & 0 & & & . & 1 & 2 \\ & & & & & 1 & 1 \\ & & & & & & 1 \end{vmatrix} . \qquad (8)$$

Thus we see that the largest element of T^{-1} is in the (1,n)-th position. Incidentally, this will also be discovered by computing the singular vectors x_n and y_n. Therefore, to produce a small $u_{n,n}$, the (n,1)-th element of A should be permuted to the (n,n)-th position. We can do this by simply switching the first and the last column of A to produce:

$$
PAQ = \left| \begin{array}{ccccccc}
-1 & -1 & . & . & . & -1 & 1 \\
-1 & 1 & -1 & & . & -1 & 0 \\
. & & 1 & & . & . & . \\
. & & & & . & -1 & . \\
-1 & . & . & . & . & 1 & 0 \\
 & 1 & 0 & 0 & . & . 0 & 0
\end{array} \right|
= \left| \begin{array}{cc}
S & q \\
 & \\
p^T & d
\end{array} \right| .
$$

The LU-factorization of the principal submatrix S of PAQ of dimension n-1 can easily be found and by the construction outlined in the proof of Theorem 4, the following factorization of PAQ is obtained:[4]

$$
PAQ = \left| \begin{array}{cccccccc}
1 & 0 & . & . & . & . & 0 \\
1 & 2 & 0 & . & . & . & . \\
. & 1 & 2 & . & . & . & . \\
. & & . & . & 0 & . & \\
1 & . & . & 1 & 2 & 0 & \\
-1 & -1 & . & . & -1 & 1 &
\end{array} \right|
\left| \begin{array}{ccccccc}
-1 & -1 & . & . & . & -1 & 1 \\
0 & 1 & 1 & . & 1 & -1 & \\
. & . & 1 & . & . & -2^{-1} & \\
. & & & . & 1 & . & \\
. & & & . & 1 & -2^{-(n-3)} & \\
0 & . & . & . & . & 0 & 2^{-(n-2)}
\end{array} \right| .
$$

7. Numerical Experiments

The example in the last section was computed *algebraically*. In this section, we present results of some *numerical* experiments with two well-known matrices that are nearly singular but for which the usual partial pivoting strategy fails to produce *any* small pivots. All computations were performed on a DEC-2060, with a 27-bit mantissa, corresponding to a relative machine precision of approximately $.4 \times 10^{-8}$. All LU-factorizations are computed by the LINPACK [8] routine SGECO which equilibrates the matrix by scaling and uses the partial pivoting strategy. It also returns an estimate of the condition number in the l_1-norm. We note that we did *not* force the pivoting strategy of SGECO into *not* choosing the last row of PAQ as the pivoting row.

The first example is the matrix T treated in the last section, with $n = 20$. Note that from (8), we see that the size of the elements of the last column of T^{-1} *decreases* rapidly from 2^{n-2} in the (1,n)-th position to 1 in the (n,n)-th position. To verify the results of Theorem 14, we computed a sequence

---

[4]It is easier to present this *unscaled* form of the factorization.

of LU-factorizations of the matrices TQ_k, where Q_k switches the k-th column of T with the last column of T. In Table 7-1, we tabulate the value of the computed $u_{n,n}$ as a function of k. By Theorem 14, the *exact* value for $u_{n,n}$ should be equal to $m^{-1}_{k,n}$. From the table, we see that the computed $u_{n,n}$'s are exactly as predicted by Theorem 14.

The second example is a matrix W quoted by Wilkinson ([21], p. 308 and p. 325) as an example of a nearly singular matrix for which PP does not produce any small pivot. The matrix W arises in the inverse iteration with the largest eigenvalue $\lambda = 10.7461942$ of the following matrix:

$$W_{21} = \left|\begin{array}{ccccccccc} 10 & 1 & & & & & & & \\ 1 & 9 & 1 & & & & & & \\ & 1 & 8 & 1 & & & 0 & & \\ & & 1 & \cdot & \cdot & & & & \\ & & & \cdot & 0 & \cdot & & & \\ & & & & \cdot & -1 & \cdot & & \\ & 0 & & & & \cdot & -2 & 1 & \\ & & & & & & \cdot & \cdot & \\ & & & & & & & 1 & -10 \end{array}\right| .$$

Since the matrix W is *symmetric*, the eigenvector corresponding to this eigenvalue is equal to the left and right singular vectors x_n and y_n of W. Wilkinson gave the computed eigenvector x_n which turns out to have a very skew distribution of the size of its components, with $(x_n)_1$ being the largest element. Thus, according to our theory, we should permute the (1,1)-th element of W to the (n,n)-th position in order to produce a small $u_{n,n}$. To accomplish this, we simply switched the first and last row followed by switching the first and last column of W. The resulting matrix, denoted by $\widetilde{W}$, is given to SGECO. No interchanges were needed in the subsequent elimination. The pivots are tabulated in Table 7-2, together with those produced by SGECO for W and the last row of $\widetilde{W}^{-1}$. We see from the table that the last pivot $u_{n,n}$ is as small as the reciprocal of the estimated condition number. Moreover, $u_{n,n}$ is exactly equal to the reciprocal of $(\widetilde{W}^{-1})_{n,n}$, verifying Theorem 14. The array IPVT(i) listed is the pivot sequence used by SGECO for W. We see that there were interchanges up to the 10-th step, after which there were no more interchanges. This is slightly different from the results reported by Wilkinson.

Table 7-1: Computed $u_{n,n}$ of TQ_k as a function of k

Reciprocal of the Estimated $\kappa(T)$ = 0.14305115E-06

k	$u_{n,n}$
1	0.38146973E-05
2	0.76293945E-05
3	0.15258789E-04
4	0.30517578E-04
5	0.61035156E-04
6	0.12207031E-03
7	0.24414063E-03
8	0.48828125E-03
9	0.97656250E-03
10	0.19531250E-02
11	0.39062500E-02
12	0.78125000E-02
13	0.15625000E-01
14	0.31250000E-01
15	0.62500000E-01
16	0.12500000E+00
17	0.25000000E+00
18	0.50000000E+00
19	0.10000000E+01
20	0.10000000E+01

Table 7-2: Computed pivots for W and $\widetilde{W}$

Reciprocal of estimated $\kappa(W)$ = 0.2817750E-08

i	IPVT	$u_{i,i}$ of W	$u_{i,i}$ of $\bar{W}$	$(\bar{W}^{-1})_{n,i}$
1	2	0.100000000E+01	-0.207461940E+02	-0.728823240E-12
2	3	0.100000000E+01	-0.174619420E+01	-0.763179210E+07
3	4	0.100000000E+01	-0.217352030E+01	-0.309896880E+07
4	5	0.100000000E+01	-0.328611110E+01	-0.878578200E+06
5	6	0.100000000E+01	-0.444188310E+01	-0.192355800E+06
6	7	0.100000000E+01	-0.552106450E+01	-0.343797880E+05
7	8	0.100000000E+01	-0.656506970E+01	-0.519713600E+04
8	9	0.100000000E+01	-0.759387300E+01	-0.681101360E+03
9	10	0.100000000E+01	-0.861450910E+01	-0.788075260E+02
10	11	0.100000000E+01	-0.963011100E+01	-0.816456100E+01
11	11	-0.136637790E+01	-0.106423530E+02	-0.765871480E+00
12	12	-0.116522300E+02	-0.116522300E+02	-0.656426360E-01
13	13	-0.126603740E+02	-0.126603740E+02	-0.517968900E-02
14	14	-0.136672080E+02	-0.136672080E+02	-0.378685230E-03
15	15	-0.146730260E+02	-0.146730260E+02	-0.257917020E-04
16	16	-0.156780420E+02	-0.156780420E+02	-0.164422070E-05
17	17	-0.166824110E+02	-0.166824110E+02	-0.985172480E-07
18	18	-0.176862510E+02	-0.176862510E+02	-0.556824110E-08
19	19	-0.186896530E+02	-0.186896530E+02	-0.297839730E-09
20	20	-0.196926890E+02	-0.196444870E+02	-0.151203090E-10
21	21	-0.206954140E+02	-0.977744340E-07	-0.102276230E+08

8. Conclusion

In this paper, we have developed a *theory* for LU-factorizations with a small n-th pivot. Moreover, we provided the basis for practical *algorithms* for computing such factorizations. We have demonstrated the effectiveness of both the theory and the algorithms by applying them to two well-known "counter-examples" to the theme of LU-factorizations with small pivots. Although Algorithm SP is quite efficient and practical for non-pathological nearly singular matrices, we do not claim that it is the most efficient implementation of our theory. We only hope that it provides a basis for further development. The best algorithm is perhaps one that will be *guaranteed* to produce a small pivot in no more cost (in both time and space) than Gaussian Elimination, possibly using some *adaptive* pivoting strategies that estimates the colorings represented by the C_i's. Furthermore, for problems with special structures, it is important to work within the constraints imposed by the structures. We hope to address this issue in a forthcoming paper.

9. Appendix

In this appendix, we shall prove Lemma 6. Part (a) of Lemma 6 is well-known. We shall prove Part (b) only. We include the proof here because we have not been able to locate either the result or the proof in the literature.

First, we need a result on the determinant of a rank-2 modification of the identity matrix.

Lemma 24: $\det(I + uv^T + wz^T) = 1 + v^Tu + z^Tw + (v^Tu)(z^Tw) - (v^Tw)(z^Tu)$.

Proof: We shall get the determinant through the eigenvalues. The rank-2 modification has a range spanned by the vectors u and w and, in this subspace, can be represented by the matrix:

$$R_2 = \begin{vmatrix} v^Tu & v^Tw \\ z^Tu & z^Tw \end{vmatrix} .$$

Since the rank-2 modification only changes *two* of the eigenvalues of the identity matrix, the determinant of the rank-2 modification of the identity matrix can be easily expressed in terms of the eigenvalues λ_1 and λ_2 of R_2 as : $\det(I + uv^T + wz^T) = (1 + \lambda_1)(1 + \lambda_2) = 1 + (\lambda_1 + \lambda_2) + \lambda_1\lambda_2$. From the characteristic polynomial for R_2, we obtain : $\lambda_1 + \lambda_2 =$

$v^Tu + z^Tw$ and $\lambda_1\lambda_2 = (v^Tu)(z^Tw) - (v^Tw)(z^Tu)$, from which the lemma follows.

Now we can prove Part (b) of Lemma 6.

Proof: (of Part (b) of Lemma 6). With $D_1 \equiv \text{Diagonal}\{d_1, \ldots, d_{n-1}, 1\}$ and $I_0 \equiv \text{Diagonal}\{1, \ldots, 1, 0\}$, we can write $D + vw^T$ as

$$D + vw^T = D_1 \left(I_0 + (D_0^{-1}v)w^T \right)$$

$$= D_1 \left(I + (D_0^{-1}v)w^T - e_n e_n^T \right).$$

The second term on the right hand side is a rank-2 modification of the identity matrix and the result of Part (b), Lemma 6 follows by applying the result of Lemma 24.

I acknowledge helpful discussions with Drs. Youcef Saad and Stanley Eisenstat on the results in this Appendix.

References

[1] T.F. Chan.
An Improved Algorithm for Computing the Singular Value Decomposition.
ACM TOMS 8(1):?, 1982.

[2] T.F. Chan.
Deflation Techniques and Block-Elimination Algorithms for Solving Bordered Singular Systems.
Technical Report 226, Yale Computer Science Department, New Haven, CT06520, 1982.

[3] T.F. Chan.
Deflated Decomposition of Solutions of Nearly Singular Systems.
Technical Report 225, Yale Computer Science Department, New Haven, CT06520, 1982.

[4] T.F. Chan and H.B. Keller.
Arclength Continuation and Multi-Grid Techniques for Nonlinear Eigenvalue Problems.
1982.
to appear in SIAM J. Sci. Stat. Comp., June, 1982.

[5] A.K. Cline.
An Elimination Method for the Solution of Linear Least Squares Problems.
SIAM J. Numer. Anal. 10:283-289, 1973.

[6] A.K. Cline and R.J. Plemmons.
l_2-Solutions to Underdetermined Linear Systems.
SIAM Review 18:92-106, 1976.

[7] A.K. Cline, C.B. Moler, G.W. Stewart and J.H. Wilkinson.
An Estimate for the Condition Number of a Matrix.
SIAM J. Numer. Anal. 16:368-375, 1979.

[8] J.J. Dongarra, J.R. Bunch, C.B. Moler and G.W. Stewart.
LINPACK User's Guide.
SIAM, Philadelphia, 1979.

[9] G.H. Golub and C. Reinsch.
Singular Value Decomposition and Least Squares Solutions.
Numer. Math. 14:403-420, 1970.

[10] G.H. Golub and J.H. Wilkinson.
Ill-Conditioned Eigensystems and the Computation of Jordan Canonical Form.
SIAM Review 18(4):578-619, 1976.

[11] H.B.Keller.
Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems.
In P. Rabinowitz (editor), *Applications of Bifurcation Theory*, pages 359-384. Academic Press, New York, 1977.

[12] H.B. Keller.
Singular Systems, Inverse Iteration and Least Squares.
1978.
unpublished manuscript, Applied Mathematics Department, Caltech, Pasadena, California, 1978.

[13] H.B. Keller.
Numerical Continuation Methods.
1981.
Short Course Lecture Notes, National Bureau of Standards, Center for Applied Mathematics, 1981.

[14] R.G. Melhem and W.C. Rheinboldt.
A Comparison of Methods for Determining Turning Points of Nonlinear Equations.
Technical Report ICMA-82-32, Institute for Computational Mathematics and Applications, Department of Mathematics and Statistics, University of Pittsburg, Pittsburg, 1982.

[15] D. P. O'leary.
Estimating Matrix Condition Numbers.
SIAM J. Sci. Stat. Comp. 1:205-209, 1980.

[16] G. Peters and J.H. Wilkinson.
The Least Squares Problem and Pseudo-Inverses.
The Computer Journal 13(3):309-316, 1970.

[17] R.J. Plemmons.
Linear Least Squares by Elimination and MGS.
J. ACM 21:581-585, 1974.

[18] W.C. Rheinboldt.
Numerical Methods for a Class of Finite Dimensional Bifurcation Problems.
SIAM J. of Numer. Anal. 15(1):1-11, 1978.

[19] G.W. Stewart.
On the Implicit Deflation of Nearly Singular Systems of Linear Equations.
SIAM J. Sci. Stat. Comp. 2(2):136-140, 1981.

[20] R.K.H. Szeto.
The Flow Between Rotating Coaxial Disks.
PhD thesis, California Institute of Technology, 1978.

[21] J.H. Wilkinson.
The Algebraic Eigenvalue Problem.
Oxford University Press, London, 1965.