SIAM J. ALG. DISC. METH.
Vol. 7, No. 4, October 1986

© 1986 Society for Industrial and Applied Mathematics
003

THE NULL SPACE PROBLEM I. COMPLEXITY*

THOMAS F. COLEMAN† AND ALEX POTHEN‡

Abstract. The Null Space Problem (NSP) is the following: Given a $t \times n$ matrix A with $t < n$, find a sparsest basis for its null space (a *null basis*). We show that columns in a sparsest null basis correspond to minimal dependent sets of columns of A. Sparsest null bases are characterized by a greedy algorithm that augments a partial basis by a sparsest null vector. Despite this result, (NSP) is NP-hard since finding a sparsest null vector of A is NP-complete. We prove that the related problem of finding a sparsest null basis with an embedded identity matrix is NP-hard too. Finally, we study the zero-nonzero structure of sparsest null bases.

Key words. null basis, null space, sparse matrix, bipartite graph, matching, matroids, conformal decomposition

AMS(MOS) subject classifications. 05, 15, 49, 65, 68

Downloaded 01/04/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

1. Introduction and overview. The development of practical algorithms for the Linear Equality Problem (LEP) is a fundamental concern in numerical optimization. (LEP) can be expressed as

$$\text{minimize } f(x)$$
$$\text{subject to } Ax = b.$$

Here $f(x)$ is a nonlinear "objective" function $f\colon \mathbb{R}^n \to \mathbb{R}$, and we assume that f is twice-continuously differentiable. The matrix A has t rows and n columns, $t < n$, and rank $(A) = r$.

Efficient algorithms to solve this problem are needed for two reasons: First, (LEP)s result from mathematical models of several practical optimization problems. Second, (LEP)s occur as subproblems of more general optimization problems. Nonlinearly constrained optimization problems are often solved by linearizing the constraints and solving a succession of resulting (LEP)s. Thus the generalized gradient method, the augmented Lagrangian method, and the projected Lagrangian method to solve these problems are based on efficient algorithms to solve (LEP)s.

One strategy for solving (LEP), the *null space method*, involves two phases: In phase 1, a "feasible" vector y is determined that satisfies $Ay = b$. In phase 2, y is corrected by a vector z in the null space of A that decreases the value of f; that is, $Az = 0$, and $f(y+z) < f(y)$. We set $y := y + z$, and repeat phase 2 until f is small enough in value, or no further reduction in its value can be made.

The correction z can often be chosen so that the algorithm converges at a quadratic rate to a stationary point of f. Let N be a basis for the $(n-r)$-dimensional null space of A (a null basis), $g(y) \in \mathbb{R}^n$ the gradient of f at y, and $H(y) \in \mathbb{R}^{n \times n}$ the Hessian matrix of f at y. We model f about the point y by a quadratic function, and choose $y + z$ to be the minimizer of this model function. This results in the system of equations

$$N^T H(y) N p = -N^T g(y),$$

* Received by the editors July 30, 1984, and in final form November 1, 1985. This research was supported in part by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under contract DE-AC02-83-ER10369, and a Cornell fellowship to the second author. This work was presented at the SIAM National Meeting at Denver, Colorado June 1983, and at the SIAM Conference on Numerical Optimization at Boulder, Colorado in June 1984.

† Department of Computer Science, Cornell University, Ithaca, New York 14853.

‡ Center for Applied Mathematics, Cornell University, Ithaca, New York 14853. Present address Department of Computer Science, The Pennsylvania State University, University Park, Pennsylvania 16802.

Downloaded 01/04/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

which is solved for the vector p, and then the correction z is computed from the equation $z = Np$.

The system of equations may be solved by computing a factorization of the projected Hessian N^THN when $n-r$ is small. For problems where $n-r$ is large, an iterative technique such as the conjugate gradient method may be used. This is a simplified discussion which ignores several practical issues; Gill, Murray, and Wright (1981) contains a more detailed discussion of (LEP).

Our concern will be with large-scale (LEP). In such problems, the constraint matrix A has a large number of rows and columns. Fortunately, however, most of the matrix elements of A are usually zeros and do not need to be stored. This redeeming feature results from each equation being involved with only a few variables, and each variable occurring only in a small number of equations. Only nonzero elements are stored, allowing large matrices to be processed without exceeding storage capacities of computers. Such matrices, whose zero-nonzero structure can be used to advantage, are *sparse.* Coleman (1984) discusses the various issues that arise in large sparse numerical optimization.

Sparsity in A is good, but is not enough. The null space algorithm needs a representation of a null basis N of A. Such a basis, being a set of $n-r$ vectors that span the null space of A, is not unique, and care needs to be taken to make it as sparse as possible.

With the above discussion to motivate us, we study the Sparse Null Space Basis Problem:

(NSP) A $t \times n$ matrix A with $t < n$ and rank r is given. Find a matrix N with the fewest nonzeros, whose columns span the null space of A.

Hereafter we will abbreviate this to the Null Space Problem. Such an $n \times (n-r)$ matrix N is a *sparsest null basis.*

This paper has four additional sections. We characterize sparsest null bases in § 2 by means of conformal decompositions and matroid theory. The computational complexity of (NSP) and some variants are discussed in § 3. The zero-nonzero structure of sparsest null bases is studied in § 4. In the last section we summarize our results, discuss related work by other researchers, and indicate future research directions. We adopt the notational convention that a term is in *italic* font when it is being defined.

In a second paper, Coleman and Pothen (1985), we will describe our algorithms for computing sparse null bases. These algorithms have two phases: in the first combinatorial phase, a maximum matching in the bipartite graph of A is used to identify the nonzero elements in the null basis. In the second numeric phase, systems of equations are solved to compute numerical values of the nonzeros in the basis. This two-phase strategy makes it possible to efficiently compute sparse null bases. Our computational experience with these algorithms will also be included.

2. A characterization of sparsest null bases.

In this section we characterize sparsest null bases by means of a "greedy" algorithm which chooses, at each step, a sparsest possible null vector to be in the basis.

An important concept in what follows is that of a circuit. A linearly dependent set of columns of the matrix A will be called a *dependent set.* A null vector of the matrix A can be obtained from the coefficients of the linear combination. A *circuit* C is a minimal dependent set—i.e., C is dependent, but all proper subsets of C are linearly independent. We will call the null vector associated with a minimal dependent set also a circuit.

Downloaded 01/04/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

ALGORITHM 2.1 (*Greedy Algorithm*). Given a $t \times n$ matrix A with $\operatorname{rank}(A) = r$, find a sparsest null basis N.

```
$N := \emptyset$
**for** $i = 1, \ldots, n - r \rightarrow$
    find a sparsest null vector $n_i$
    such that rank $(n_1, \ldots, n_i) = i$.
    $N := N \cup n_i$ **rof**.
```

THEOREM 2.1 (Optimality Theorem). *The matrix N is a sparsest null basis of A if and only if it can be constructed by the greedy algorithm.*

Algorithm 2.1 is greedy, since it augments the partial null basis at each step by a sparsest null vector linearly independent of those previously chosen. To us Theorem 2.1 is a surprising result; locally greedy strategies seldom lead to globally optimal solutions to optimization problems. We now develop the results needed for its proof.

Let the jth component of a vector x be denoted by $(x)_j$. (This should not be confused with the notation for a vector, say n_i.) We define the *support* of x, $S(x)$, to be

$$S(x) = \{j: (x)_j \neq 0\}.$$

By definition, if c is a circuit, there cannot exist a null vector x with $S(x) \subset S(c)$.

LEMMA 2.2. If c, d are circuits of A, and $S(c) = S(d)$, then c is a scalar multiple of d.

Proof. Suppose the lemma is false. Then we can pick a scalar λ such that $(c)_i - \lambda(d)_i = 0$, for some $i \in S(c)$. But then $S(c - \lambda d) \subset S(c)$, and c is not minimal. □

Hence circuits of A are unique to within a multiplicative constant. We now introduce a linear algebraic concept from network flow theory, conformal decomposition, studied first by Camion (1968), Fulkerson (1968), and Rockefellar (1969). Lemmas 2.2 through 2.4 follow immediately from their work.

A vector x *conforms* to a vector y if

$$(x)_j \neq 0 \Rightarrow ((y)_j \neq 0, \text{ and } \operatorname{sgn}\{(x)_j\} = \operatorname{sgn}\{(y)_j\})$$

where sgn denotes the sign function. For example, let

$$\operatorname{sgn}(x) = (+0-0+0), \qquad \operatorname{sgn}(y) = (++-0+-),$$

then x conforms to y, but y does not conform to x. Note that if x conforms to y, then $S(x) \subseteq S(y)$.

LEMMA 2.3. *Given a null vector n, there exists a circuit c that conforms to it.*

Proof. Again, the proof is by contradiction. Choose a null vector x with the smallest $|S(x)|$ such that no circuit of A conforms to it. Let c be a circuit with $S(c) \subset S(x)$. Define the set

$$J = \{j: (c)_j \neq 0, \text{and } (c)_j \text{ and } (x)_j \text{ disagree in sign}\}.$$

J is not the empty set, else c would conform to x. Let

$$a = \min_{j \in J} -\frac{(x)_j}{(c)_j}.$$

Consider the vector $z = x + ac$. By construction, z conforms to x, and $S(z) \subset S(x)$. By the selection of x there is a circuit d that conforms to z. But then d conforms to x. □

We can now apply Lemma 2.3 repeatedly to get

Downloaded 01/04/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

LEMMA 2.4. *A null vector x can be expanded in a sum of distinct circuits*

$$x = c_1 + \cdots + c_p,$$

where each circuit c_i conforms to x.

The above expansion is the *conformal decomposition* of a null vector of A; it is not necessarily unique. A more general decomposition exists for a vector of any subspace of $\mathbb{R}^n$, and is discussed by Camion, Fulkerson and Rockefellar. We can now use Lemma 2.4 to prove that we need concern ourselves only with circuits to solve (NSP).

THEOREM 2.5. *Each sparsest null vector n_i chosen by the greedy algorithm is a circuit.*

Proof. The proof is by induction on i. The result is clearly true for n_1. By the inductive hypothesis, assume that the theorem is true for all n_j, where $1 \leq j < i$.

Suppose that n_i is not a circuit. Conformally decompose n_i into a sum of circuits. At least one of the circuits in this sum, say c, must be linearly independent of $(n_1, \cdots, n_{i-1})$ since n_i is independent of them. Since n_i is not a circuit, $S(c) \subset S(n_i)$, and c is a sparser null vector than n_i which the algorithm could have chosen at this step. □

A similar argument can be used to prove

THEOREM 2.6. *Each column of a sparsest null basis N is a circuit.*

Theorem 2.6 states that the only dependent sets of interest in (NSP) are circuits. Since the greedy algorithm chooses only circuits by Theorem 2.5, the possibility now looms that it could find a sparsest null basis. As Theorem 2.1 states, this suspicion is correct; and a stronger result holds, namely, every sparsest null basis can be found by the greedy algorithm.

We now introduce the matroid concepts used to prove Theorem 2.1. Let E be a finite set. Some of the subsets of E are defined to be *independent*; a subset of E that is not independent is *dependent*. Let

$$H = \{I \subseteq E: I \text{ is independent}\}.$$

We consider the situation when the independent sets satisfy the following two properties:

(M1) All subsets of an independent set are independent. (The empty set is independent by this property if H is not empty.)

(M2)- Let I_p and I_{p+1} be independent sets with p and $p+1$ elements respectively. Then there is an element $e \in I_{p+1} \backslash I_p$ such that $I_p + e$ is independent.

Let the family of independent sets H satisfy (M1) and (M2). Then the tuple $M = (C, H)$ is defined to be a *matroid* (Welsh (1976)).

The reader may find it convenient to think of E as the set of columns of a matrix. An independent subset of E has linearly independent columns. By linear algebra, one can establish that both (M1) and (M2) hold. Hence M is a matroid, and we call it the matroid generated by the columns of the matrix.

A minimal dependent set of a matroid is called a *circuit.* Thus far we have used the word circuit to denote a minimal linearly dependent set of columns of a matrix. This usage is consistent with the definition of a circuit of a matroid. What we call a circuit of a matrix is indeed a circuit of the matroid generated by the columns of the matrix.

A *maximal independent set* is an independent set all supersets of which are dependent. We call such a set a *basis* of M. Every basis of M has the same size, which is called its *rank.*

Proof of Theorem 2.1. By Theorems 2.5 and 2.6 we can restrict our attention to circuits of A. Since A has n columns, it has only a finite number of circuits. Let C be

Downloaded 01/04/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

the circuit matrix whose columns are all the circuits of A. Thus

$$C=(c_1, \cdots, c_q).$$

Let M be the matroid generated by the columns of C. To each circuit c_i, assign the positive integer weight $|S(c_i)|$. Algorithm 2.1 is equivalent to choosing a basis of minimum weight for the circuit matroid M. Theorem 2.1 now follows from two well-known results on matroids:

(1) The matroid greedy algorithm constructs a basis of minimum weight.

(2) The weight of the kth smallest element of such a basis is no bigger than the kth smallest element of any other independent set (Lawler (1976)). □

Unfortunately, the proof of Theorem 2.1 does not lead immediately to a polynomial time algorithm to solve (NSP). The difficulty is that a matrix A of n columns and t rows might have $O(n^t)$ circuits.

3. The complexity of (NSP) and its variants. In the previous section, we showed that a sparsest null basis can be constructed by a greedy algorithm. Hence we consider the following strategy to solve (NSP): design a polynomial time algorithm for one step of the greedy algorithm. This latter algorithm would choose a sparsest circuit linearly independent of circuits chosen in previous steps. If we could design such an algorithm, then $n-r$ applications of it to the matrix A will solve (NSP).

Unfortunately, such a happy prospect is unlikely; we now discuss the reason why. The greedy algorithm chooses a circuit of minimum cardinality in its first step. We call such a circuit a *minimum circuit.* Theorem 3.1 states that the minimum circuit problem is NP-complete. Hence it is as hard as any of the problems in the class NP. For the reader unfamiliar with this terrain, Garey and Johnson (1979) is an excellent introduction to the theory of NP-completeness. Theorems 3.1 and 3.2 were proved independently by L. J. Stockmeyer, and his proofs may be found in McCormick (1983).

THEOREM 3.1 (Minimum Circuit Theorem). *Given a positive integer k, it is* NP-*complete to find a circuit of A of cardinality k or less.*

We omit our proof since our reduction is similar to Stockmeyer's. Theorem 3.1 leads to an easy proof that (NSP) is NP-hard. We do not know if (NSP) is in NP.

THEOREM 3.2 (Sparsest Null Basis Theorem). *Given a positive integer k, it is* NP-*hard to find a null basis of A with k or fewer nonzeros.*

Proof. By Theorem 2.1, every sparsest null basis contains a minimum circuit. By Theorem 3.1, it is NP-complete to find a minimum circuit. □

If A is restricted to be the vertex-edge incidence matrix of a graph $G=(V, E)$, a minimum circuit can be found in $O(|V|\,|E|)$ time by an algorithm of Itai and Rodeh (1978). In this situation, a minimum circuit corresponds to a cycle in the graph with the minimum number of edges. Matroids generated by vertex-edge incidence matrices of graphs are called *graphic matroids.*

Every matroid has a dual defined on the same ground set C. A basis of the dual matroid is the complement of a basis of the primal matroid. A matroid dual to a graphic matroid is *cographic.* Minimum circuits of cographic matroids correspond to minimum cuts in the graph; these can also be found in polynomial time.

A matrix A is *totally unimodular* if every subdeterminant of A is either $+1$, -1, or 0. The matroid generated by such a matrix is called a *totally unimodular matroid.* Seymour (1980) has shown that any totally unimodular matroid can be decomposed by a polynomial time algorithm into a matroid sum of graphic matroids, cographic matroids, and copies of a special matroid on ten elements. It follows that minimum circuits of totally unimodular matroids can be determined in polynomial time.

Downloaded 01/04/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

We have shown that constructing a null basis of a matrix A with the maximum number of nonzeros is also NP-hard when the columns in the basis are circuits of A.

THEOREM 3.3. *Given a positive integer k, it is* NP-*hard to find null basis of A with k or more nonzeros, if each column in the basis is a circuit.*

The proof is by the restriction of A to vertex-edge incidence matrices of graphs, and uses the result that finding a basis with the maximum number of edges for the cycle space of a graph is NP-complete. A proof is presented in Pothen (1984).

Since (NSP) is NP-hard, we cannot expect to construct sparsest null bases by a polynomial time algorithm. Hence we lower our sights in terms of sparsity, and ask how hard it is to construct a sparsest null basis with a prescribed zero-nonzero structure.

Current null space algorithms for (LEP) use the variable-reduction technique proposed by Wolfe (1962) to construct null bases. Let A_r denote any r linearly independent rows of A. The matrix A_r is partitioned (after possible column permutations) as

$$A_r = (M \quad U),$$

where M is a $r \times r$ nonsingular matrix. Then we construct the matrix

$$N = \begin{pmatrix} -M^{-1}U \\ I_{n-r} \end{pmatrix},$$

where I_{n-r} is the identity matrix of dimension $n-r$. Since $AN = 0$, the columns of N are null vectors of A. Each of the last $n-r$ rows of N has only one nonzero in it, and so linear combinations of the columns of N cannot produce the zero vector. Hence N is a null basis. We call a basis with an embedded identity submatrix a *fundamental null basis.*

We formally state the Fundamental Null Space Problem:

(FNSP) Given a $t \times n$ matrix A of rank r and a positive integer k, find a fundamental null basis N with k or fewer nonzeros.

THEOREM 3.4. (FNSP) *is* NP-*hard.*

The proof of this theorem uses a result on spanning trees of graphs. We now develop the concepts needed for the proof.

Let $G = (V, E)$ be a connected graph on ν vertices and ε edges with vertex-edge incidence matrix $M(G)$. A *cycle* in G is a sequence of distinct vertices $v_1, \ldots, v_{k-1}, v_k \equiv v_1$, where $(v_{i-1}, v_i) \in E$ for $i = 2, \ldots, k$. Denote the edge incidence vector of a cycle by Γ, with component γ_i equal to 1 if e_i is an edge of the cycle, and 0 otherwise. Since each vertex in the cycle is an endpoint of exactly two edges, we have

$$M(G)\Gamma^T = 0,$$

over the binary field GF (2). Thus Γ is a null vector of $M(G)$. Further, since the omission of any edge in the cycle will violate the equation, Γ is a circuit of $M(G)$. Since every circuit of M has zero or two edges incident on each vertex of G, there is a one-to-one correspondence between a cycle of G and a circuit of $M(G)$ over GF (2).

However, our interest is with circuits over the real field. But the restriction to arithmetic over GF(2) is easily removed. Let D be a directed graph obtained by arbitrarily directing the edges of G. The vertex-edge incidence matrix of D, $M(D)$ has in the column of the directed edge $\{u, v\}$ the entry $+1$ in the row of v, -1 in the row of u, and 0 in all other rows. A *cycle* in D is defined to be a cycle in G with an arbitrary orientation. Let $\Gamma(D)$ be the edge incidence vector of a cycle in D, with component γ_i equal to $+1$ if e_i is an edge in the cycle and the orientations of the cycle and e_i

Downloaded 01/04/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

agree, -1 if e_i is an edge in the cycle and the orientations disagree, and 0 if e_i is not an edge in the cycle. We have

$$M(D)\Gamma(D)^T = 0,$$

where the arithmetic is now over the real field. Further, there is a one-to-one correspondence between a cycle of D and a circuit of $M(D)$.

We now extend this correspondence to one between a fundamental null basis of $M(D)$ and an appropriate graph concept. A *spanning tree* T of an undirected graph G is a connected subgraph with ν vertices and $\nu - 1$ edges. Each nontree edge e creates a unique cycle $C(T, e)$ in the subgraph $T + e$. We call $C(T, e)$ the *fundamental cycle* created by e with respect to T. Since T has $\nu - 1$ edges, there are $\omega(G) \equiv \varepsilon - \nu + 1$ nontree edges. Hence there are $\omega(G)$ fundamental cycles with respect to T. *The fundamental cycle matrix* $\Phi(G)$ has $\omega(G)$ rows and ε columns, with element ϕ_{ij} equal to 1 if e_j is an edge of the cycle Φ_i, and 0 otherwise. If the edges of T are numbered from 1 to $\nu - 1$, and the nontree edges from ν to ε, then $\Phi(G)$ has the structure $\Phi(G) = (\Phi_{11} \ \ I)$.

Let D be a directed graph obtained from G as before. A *spanning tree* of D is defined to be a spanning tree of G. The *fundamental cycle matrix of* D, $\Phi(D)$, has element ϕ_{ij} equal to $+1$ if e_j is an edge of Φ_i and their orientations agree, -1 if e_j is an edge of Φ_i and their orientations disagree, and 0 if e_j is not an edge of Φ_i. Thus for any spanning tree T, $\Phi(G)$ and $\Phi(D)$ have the same structure. Since each row of $\Phi(D)$ corresponds to a cycle in D, we have

$$M(D)\Phi(D)^T = 0,$$

where the arithmetic is over the real field. Hence $\Phi(D)^T$ is a fundamental null basis for $M(D)$.

Proof of Theorem 3.4. Restrict A to vertex-edge incidence matrices of directed graphs. A sparsest fundamental null basis of A now corresponds to a sparsest fundamental cycle matrix of the associated directed graph. The latter is equivalent to finding a sparsest fundamental cycle matrix of the undirected graph obtained by ignoring the directions of the edges. This last problem is that of finding a spanning-tree that minimizes the total number of edges in the set of fundamental cycles with respect to it. This problem is NP-complete; proofs may be found in Deo, Prabhu, and Krishnamoorthy (1982) and Pothen (1984). □

A fundamental basis for the row space of A has the structure $(I_r \ \ B)$ and corresponds to a fundamental null basis

$$\begin{pmatrix} -B \\ I_{n-r} \end{pmatrix}$$

with only a constant change in the number of nonzeros. Hence we have

COROLLARY 3.5. *Given a positive integer k, it is* NP-*hard to find a fundamental row space basis of A with k or fewer nonzeros.*

In contrast, finding a (nonfundamental) sparsest row space basis can be done in polynomial time (Hoffman and McCormick (1984)) when the matrix A satisfies a nondegeneracy assumption called the matching property.

4. The structure of sparsest null bases.

Any algorithm for constructing a null basis has to ensure that the set of $n - r$ null vectors chosen is linearly independent. Constructing a fundamental null basis makes this easy to do. However, sparsest null bases need not be fundamental. We may be constrained to construct relatively dense fundamental bases where sparse nonfundamental null bases may exist.

Downloaded 01/04/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

But what zero-nonzero structure (hereafter *structure*) should a sparsest null basis have? By Theorem 4.2 below, a set of m vectors is linearly independent for all nonzero values of its nonzero elements if and only if it has an embedded upper triangular submatrix of dimension m. In what follows, let V be a matrix with n rows and m columns, with $n > m$. Distinguish some elements of V as nonzeros and the rest as zeros. By a value of a matrix we mean an assignment of nonzero numerical values to its nonzero elements.

LEMMA 4.1. *If V has at least two nonzeros in each row, then there exists a nonzero vector x, and a value for V, such that $Vx = 0$.*

Proof. Let row i have $|r_i| \geqq 2$ nonzeros. We assign to any $|r_i| - 1$ nonzeros the value$+1$, and to the remaining element the value $1 - |r_i|$. We do this for all the rows of V, and choose $x = (1 \dots 1)^T$. □

THEOREM 4.2. *V has rank m for all values if and only if it can be permuted to the following structure*:

$$V = \begin{pmatrix} B \\ U_m \end{pmatrix},$$

where U_m is an $m \times m$ upper triangular matrix with nonzero diagonal elements.

Proof. The if part is obvious. We prove the only if part. Suppose that V has rank m, but does not have the structure claimed. Permute the rows and columns of V so that it has the structure

$$V = \begin{pmatrix} B & C \\ O & R \end{pmatrix},$$

where R is upper triangular and maximal with respect to this property. Since R is maximal, B has at least two nonzeros in each of its rows. By Lemma 4.1, we can now find a vector x and numeric values for the nonzeros of B so that $Bx = 0$. Since

$$V\begin{pmatrix} x \\ 0 \end{pmatrix} = 0,$$

V does not have rank m. This contradiction proves the theorem. □

It may appear from this theorem that a sparsest null basis should have an embedded upper triangular matrix. This would be true if we could assign any value to N. But, we are not free to do so. We can assign any value to A; then, once the structure of N is chosen, the values of the columns of N are uniquely determined to within a multiplicative constant.

Theorem 4.3 concerns the structure of a sparsest null basis. This result is a matroid generalization of a theorem on cycles in graphs proved by Stepanets (1964). We shall denote the set of columns of the matrix A also by A. Let $n(a_j)$ be a circuit of minimum cardinality containing the column a_j. The reader may find Fig. 1 helpful to follow the proof of this theorem.

THEOREM 4.3 (Generalized Stepanets Theorem). *Let the columns $a_1, \cdots, a_k$ be chosen such that*

$$a_1 \in A,$$
$$a_2 \in A \backslash n(a_1), \quad \dots,$$
$$a_k \in A \Big\backslash \bigcup_{j=1}^{k-1} n(a_j).$$

There exists a sparsest null basis N among whose columns are the circuits $n(a_1), \cdots, n(a_k)$.

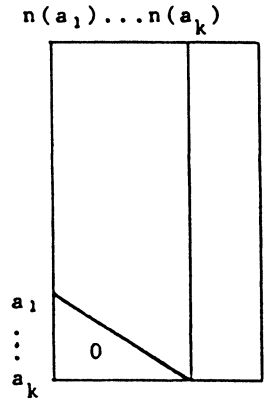

FIG. 1. *The sparsest null basis N.*

Proof. We prove the theorem by induction on $n(a_i)$. Let $q=n-r$, and denote the set $(P\backslash p)\cup n$ by $P-p+n$.

Let $P=(p_1\cdots p_q)$ be a sparsest null basis of A. Since P is a basis, we can expand $n(a_1)$ as

$$n(a_1)=c_1p_1+\cdots+c_mp_m.$$

We assume that all the coefficients in this expansion are nonzero. Of the circuits in this equation, there must exist at least one circuit, say p_h, which contains a_1. Consider the system

$$P_1=P-p_h+n(a_1).$$

Clearly P_1 is a null basis. Further, since $n(a_1)$ has minimum cardinality over circuits containing a_1, P_1 is a sparsest null basis.

For the inductive step, assume that P_{j-1} is a sparsest null basis of A, having among its columns $n(a_1),\ldots,n(a_{j-1})$, where each a_i is chosen as claimed. We choose $n(a_j)$ to be a circuit of minimum cardinality containing a_j. Expand $n(a_j)$ in the basis P_{j-1},

$$n(a_j)=c_1p_1+\cdots+c_mp_m,$$

where again each of the coefficients is nonzero. There is at least one circuit in this equation, say p_h, which contains a_j. The circuit p_h cannot be any one of $n(a_1),\cdots,n(a_{j-1})$ by the choice of a_j. Consider now the system

$$P_j=P_{j-1}-p_h+n(a_j).$$

As before, P_j is a sparsest null basis of A.

We take N to be P_k. This completes the proof. □

There is some $k\leqq n-r$ for which choosing a column a_{k+1} is not possible, since the first k circuits contain all columns of A.

COROLLARY 4.4. *If $k=n-r$ in Theorem 4.3, then the system of circuits $n(a_1),\cdots,n(a_k)$ is a sparsest null basis N of A.*

In this case, N has an upper triangular submatrix with $n-r$ columns. We call such a basis a *triangular null basis.* Thus, some sparsest null bases are triangular.

5. Conclusions. We have formulated the Null Space Problem, and the Fundamental Null Space Problem. We have shown that only circuits can be columns in a sparsest null basis, and that such a basis can be characterized by a matroid greedy

Downloaded 01/04/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 01/04/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

algorithm. However, (NSP) is NP-hard since a sparsest null basis contains a minimum circuit and finding a minimum circuit is NP-complete. Constructing a sparsest fundamental null basis is also an NP-hard problem. Hence the use of approximation algorithms to solve (NSP) is justified.

A fundamental null basis ensures linear independence of the set of null vectors chosen. We have extended this observation to show that a set of vectors is linearly independent for all values if and only if it has an embedded upper triangular submatrix with nonzeros on the diagonal. This can be used in approximation algorithms to construct triangular null bases for which linear independence of the null vectors is again easy to ensure.

A problem related to (NSP) is that of finding a set of cycles with the fewest edges that spans the cycle space of a graph. Note that our proof technique for the NP-hardness of (NSP) does not extend to this problem, since cycles with the fewest edges can be found in polynomial time. The complexity of this problem is open (Johnson (1985)). However, the problem of finding a set of fundamental cycles with the fewest edges that span the cycle space of a graph is NP-complete (Deo, Prabhu and Krishnamoorthy (1982), Pothen (1984)). The problem of finding a set of cycles with maximum number of edges spanning the cycle space of a graph is NP-complete (Pothen (1984)).

In Coleman and Pothen (1985), we will show how circuits can be constructed from a maximum matching in the bipartite graph of the matrix A. This algorithm can be used repeatedly to construct fundamental null bases. Here the sparsity of the basis turns out to depend only on the partition of the columns of A into the matched and unmatched sets. Various heuristic strategies for finding particular matchings are used to obtain sparse null bases.

By varying the matching while constructing null vectors, a triangular null basis can be obtained. Such bases can be potentially sparser than fundamental null bases; however, this increase in sparsity is achieved at greater computational cost.

We briefly mention recent work related to (NSP). Berry, Heath, Kaneko, Lawo, Plemmons and Ward (1985) have implemented a refined version of a "turnback algorithm", proposed initially by Topcu (1979), that constructs sparse null bases for large sparse, banded A. This algorithm uses an initial numeric factorization of A to identify subsets of columns that could become dependent sets in the $n-r$ null vectors. In a second turnback phase, a numeric factorization on each dependent set is performed to obtain circuits. Their numerical results on several problems arising from finite element models in structural engineering show that they obtain null bases with the same degree of sparsity as the input matrices. Berry and Plemmons (1985) have implemented a parallel version of this algorithm on a Denelcor HEP computer. Gilbert and Heath (1986) have implemented several algorithms for computing sparse null bases; some of these are closer in spirit to the ones we have designed. For instance, in one of their algorithms, they construct a triangular null basis; the columns in each circuit are identified by matching methods.

Much work remains to be done. An important numerical consideration is the condition number of the null basis. To this end, algorithms that can compromise some degree of sparsity for better conditioned null bases will need to be developed. Other sparsity criteria than the one used in this paper need to be studied. We mention one such in closing. An *implicit null basis* is a representation for the null basis as a product of a sequence of elementary matrices (e.g., Givens rotations), with the sequence of elementary matrices being stored. A sparse implicit null basis has relatively few elementary matrices in the sequence. One direction in which we plan to continue this research is in developing sparse implicit orthogonal null bases.

Acknowledgments. We wish to thank Professors John Gilbert and Bob Bland for several valuable discussions and Ms. Helen Johnston for technical assistance.

REFERENCES

M. W. BERRY, M. T. HEATH, I. KANEKO, M. LAWO, R. J. PLEMMONS AND R. C. WARD, *An Algorithm to compute a sparse basis of the null space*, Numer. Math., 47 (1985), pp. 483-504.

M. W. BERRY AND R. J. PLEMMONS, *Computing a banded basis of the null space on the Denelcor* HEP *multiprocessor*, in Proc. AMS/SIAM Summer Conference on Linear Algebra in Systems Theory, AMS Series on Contemp. Math., 1985.

P. CAMION, *Modules unimodulaires*, J. Combin. Theory, 4 (1968), pp. 301-362.

THOMAS F. COLEMAN, *Large Sparse Numerical Optimization*, Lecture Notes in Computer Science 165, Springer-Verlag, Berlin, 1984.

THOMAS F. COLEMAN AND ALEX POTHEN, *The sparse null space basis problem* II. *Algorithms*, (in preparation), 1985.

NARSINGH DEO, G. M. PRABHU AND M. S. KRISHNAMOORTHY, *Algorithms for generating fundamental cycles in a graph*, ACM Trans. Math. Software, 8 (1982), pp. 26-42.

D. R. FULKERSON, *Networks, frames, blocking systems*, in Mathematics of the Decision Sciences, Vol. 2, G. B. Dantzig and A. F. Veinott, eds, American Mathematical Society, Providence, RI, 1968, pp. 303-334.

M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of* NP-*Completeness*, W. H. Freeman, San Francisco, 1979.

JOHN R. GILBERT AND MICHAEL T. HEATH, *Computing a sparse basis for the null space*, Cornell University and Oak Ridge National Laboratory Technical Report, 1986.

PHILIP E. GILL, WALTER MURRAY AND MARGARET H. WRIGHT, *Practical Optimization*, Academic Press, New York, 1981.

ALAN J. HOFFMAN AND S. THOMAS MCCORMICK, *A fast algorithm for making matrices optimally sparse*, in Progress in Combinatorial Optimization, W. R. Pulleyblank, ed., Academic Press, New York, 1984.

ALON ITAI AND MICHAEL RODEH, *Finding a minimum circuit in a graph*, SIAM J. Comput., 7 (1978), pp. 413-423.

DAVID S. JOHNSON, *The* NP-*Completeness column: An ongoing guide*, J. Algorithms, 6 (1985), pp. 145-159.

EUGENE L. LAWLER, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart, and Winston, New York, 1976.

S. THOMAS MCCORMICK, *A combinatorial approach to some sparse matrix problems*, SOL 83-5, Ph.D. Thesis, Stanford Univ., Stanford, CA, 1983.

ALEX POTHEN, *Sparse null bases and marriage theorems*, Ph.D. Thesis, Cornell Univ. Ithaca, NY, 1984.

R. T. ROCKEFELLAR, *The elementary vectors of subspaces of* $\mathbb{R}^n$, in Combinatorial Mathematics and its Applications, R. C. Bose and T. A. Dowling, eds., Univ. North Carolina Press, Chapel Hill, NC 1969, pp. 104-127.

P. D. SEYMOUR, *Decomposition of regular matroids*, J. Combin. Theory Ser. B, 28 (1980), pp. 305-359.

G. F. STEPANETS, *Basis systems of vector cycles with extremal properties in graphs*, Upsekhi Mat. Nauk., 19 (1964), pp. 2, 171-175. (In Russian.)

A. TOPCU, *A contribution to the systematic analysis of finite element structures through the force method*, Ph.D. Thesis, Univ. of Essen, Essen, Germany, 1979. (In German.)

D. J. A. WELSH, *Matroid Theory*, Academic Press, London, 1976.

PHILIP WOLFE, *The reduced gradient method*, The RAND Corporation, 1962, unpublished manuscript.

Downloaded 01/04/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php