Stable numerical algorithms for equilibrium systems*

Stephen A. Vavasis†

September 9, 1992

Abstract

An equilibrium system (also known as a KKT system, a saddle-point system, or a sparse tableau) is a square linear system with a certain structure. G. Strang has observed that equilibrium systems arise in optimization, finite elements, structural analysis, and electrical networks. Recently, G. W. Stewart established a norm bound for a type of equilibrium system in the case that the "stiffness" portion of the system is very ill-conditioned. In this paper we investigate the algorithmic implications of Stewart's result. We show that all standard textbook algorithms for equilibrium systems are unstable. Then we show that a certain hybrid method has the right stability property.

1 Equilibrium systems

Recently, Strang [1986] has observed that the problem of solving the structured linear system

$$\begin{pmatrix} D & -A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix} \tag{1}$$

*This work supported by an NSF Presidential Young Investigator grant, with matching funds received from Xerox Corp.

†Department of Computer Science, Upson Hall, Cornell University, Ithaca, NY 14853.

for $\mathbf{x}$ and $\mathbf{y}$ arises in many physical applications. Here, D is an $m \times m$ matrix, A is an $m \times n$ matrix, $\mathbf{x}$ and $\mathbf{b}$ are m-vectors, and $\mathbf{y}$ and $\mathbf{c}$ are n-vectors. We call this system an **equilibrium system.**

The focus of this paper is on the stability of algorithms for this linear system. Here is a table summarizing the applications, and the interpretations of $D, A, \mathbf{x}, \mathbf{y}, \mathbf{b}, \mathbf{c}$ in these applications. In the case of finite elements, we have indicated the interpretations in the context of a heat equilibrium problem (Poisson's equation).

	Optimization	Electrical networks	Finite elements	Structures
D	Objective function Hessian	Resistances	Thermal resistance	Element flexibility
A	Constraints	Node-wire adjacency	Node-element geometry	Node-element geometry
$\mathbf{x}$	Primal variables	Currents	Element heat flow	Element forces
$\mathbf{y}$	Lagrange multipliers	Voltages	Nodal temperatures	Nodal displacements
$\mathbf{b}$	Objective function gradient	Voltage sources	Applied temperature gradient	Element dilations
$\mathbf{c}$	Distance from constraints	Current sources	Applied heat sources	Applied nodal forces

The reader will observe that there are some similarities among the various interpretations. For example, one similarity is that, for the three physical applications, $\mathbf{x}$ and $\mathbf{c}$ are measured in the same physical units (e.g., amps), as are $\mathbf{y}$ and $\mathbf{b}$ (e.g., volts). This is also the case in certain optimization problems, such as flow problems. This observation has some importance for numerical algorithms.

In all of these applications, the following two assumptions are commonplace, and they are made throughout the paper.

A1. *Matrix D is symmetric and positive definite.*

A2. *Matrix A has rank n, i.e., full column rank.*

These assumptions imply that (1) is a nonsingular linear system with a unique solution.

The main focus of this paper is what happens when D is severely ill-conditioned. In the case that D is well-conditioned, the numerical problems associated with solving (1) are generally not as troublesome, and most standard methods will give good answers. Thus, we make the following assumption:

A3. *Matrix D is very ill-conditioned.*

The most natural framework for this assumption is an optimization algorithm involving a barrier function. The primary example of a barrier function in optimization is the class of interior point methods for linear programming. In an interior point method, matrix D becomes very ill-conditioned when the iterate approaches the boundary of the feasible region. (See Karmarkar [1984] for the first interior-point method proposed for linear programming. See Wright [1992] for a description of barrier methods, linear programming, and their relationship.) For linear programming, since the solution is always on the boundary of the region, ill-conditioning in D *always* occurs during the algorithm.

Ill-conditioning of D can also occur in the other three applications listed above. In electrical networks, ill-conditioning occurs, for example, when one wishes to model the leakage currents to ground through insulators in the electrical network. In this case, some of the resistances will be many orders of magnitude larger than others, leading to a very ill-conditioned D. The same ill-conditioning occurs if the linear equilibrium system is a model of a time-varying system in which certain circuit elements have switched "off."

In structural analysis, ill-conditioning in D occurs when the elements of the structure have widely different rigidity properties. In the finite elements for a heat application, ill-conditioning in D would occur if part of the domain under analysis were a thermal insulator and another part were a thermal conductor.

In the case that D is severely ill-conditioned, it is not at all apparent that an accurate numerical solution to (1) is possible, since ill-conditioning in D will presumably mean extreme sensitivity to roundoff errors. A recent theorem due to Stewart [1989] shows that, in principle, an accurate solution can be computed under further assumptions. In Section 2 we review Stewart's result. Further theory is developed in Section 3. None of the standard algorithms are stable, however, when compared with the theoretical bound, as we demonstrate in Section 4. Our main contribution is to argue in Section 5 and Section 6 that a certain algorithm can in fact accurately solve (1).

Most of the analysis of this paper is valid only under certain further assumptions, which we now state.

A4. *We are more interested in recovering* $\mathbf{y}$ *in* (1) *rather than* $\mathbf{x}$.

A5. *In* (1), *the vector* $\mathbf{c}$ *is zero.*

A6. *Matrix* D *in* (1) *is a diagonal matrix.*

In Section 8 we explain how A4, A5, and A6 could be relaxed.

A4 is similar to obtaining error bounds on individual components of the solution of a linear system. See, for example, Chandrasekaran and Ipsen [1991]. In our work, the objective is to obtain a bound for a block of components of the solution.

In the context of optimization, A5 means that the current point is feasible. In the context of electrical networks, this assumption means that no external current sources are applied—only voltage sources (e.g., batteries or generators).

A6 holds in the context of interior point methods for linear programming. It also holds for electrical networks composed of batteries and resistors. In structure analysis and finite elements, this assumption holds in the case that the domain is a one-dimensional object (e.g., heat flow in a thin bar). For two- and three-dimensional objects, one usually obtains a matrix D that is block-diagonal.

A fifth application of the equilibrium system, not listed in the above table, is a discretization of Stokes flow. Stokes flow is a linear approximation to the Navier-Stokes equations for incompressible fluid flow, under the assumption of a very low Reynolds number. A6 is never satisfied for Stokes flow (even in one dimension), which is why we do not discuss it further.

2 Stewart's norm-bound result

Under the assumptions made in the last section, we can apply Stewart's theorem. Because of A4, it is useful to write an explicit formula for $\mathbf{y}$ in terms of the other variables. This formula is obtained by eliminating $\mathbf{x}$ from (1), and also substituting A5:

$$A^T D^{-1} A\mathbf{y} = -A^T D^{-1}\mathbf{b}. \tag{2}$$

Because of A1 and A2, the matrix $A^T D^{-1} A$ is invertible. Thus, this linear system uniquely determines $\mathbf{y}$:

$$\mathbf{y} = -(A^T D^{-1} A)^{-1} A^T D^{-1}\mathbf{b}. \tag{3}$$

We are now in a position to state Stewart's theorem. For the sake of completeness, we replicate his proof of the theorem, because we refer to some of the proof steps in our algorithm construction.

Theorem 1 (Stewart, 1989). *Let $\mathcal{D}$ denote the set of all positive definite $m\times m$ real diagonal matrices. Let A be an $m\times n$ real matrix of rank n. Then there exist constants χ_A and $\bar{\chi}_A$ such that for any $D\in\mathcal{D}$,*

(a) $\|(A^T D^{-1} A)^{-1} A^T D^{-1}\| \le \chi_A$, *and*

(b) $\|A(A^T D^{-1} A)^{-1} A^T D^{-1}\| \le \bar{\chi}_A$.

In this theorem, we have assumed the use of some matrix norm $\|\cdot\|$ that is induced by a vector norm (also denoted $\|\cdot\|$).

A closely related result was obtained independently by Todd [1990] as part of an analysis of Karmarkar's algorithm. In particular, Todd shows that

$$\|(A^T D^{-1} A)^{-1} A^T D^{-1}\mathbf{b}\| \le c(A, \mathbf{b}).$$

This bound can be extended to obtain Theorem 1 with a compactness argument involving $\mathbf{b}$. Todd's proof is geometric and completely different from Stewart's. We follow Stewart's proof, because it contains algebraic information useful for further development.

Before beginning the proof, we discuss the interpretation of the theorem. The matrix in (a) is (except for sign) the same matrix that is applied to $\mathbf{b}$ to obtain $\mathbf{y}$ according to (3). Thus, (a) says that $\mathbf{y}$ cannot be much larger than $\mathbf{b}$, no matter how D is chosen. In the context of electrical networks, this has a very natural physical interpretation: For a fixed electrical network with no current sources, no matter how badly the resistors are out of scale, there can never be a voltage level in the circuit that is much higher than the applied battery voltages.

Statement (b) also has an algorithmic interpretation. Specifically, if we know $\mathbf{y}$ and are trying to obtain $\mathbf{x}$, we observe from (1) that

$$D\mathbf{x} - A\mathbf{y} = \mathbf{b}.$$

The second term on the left-hand side is precisely

$$A(A^T D^{-1} A)^{-1} A^T D^{-1}\mathbf{b}.$$

Thus, the matrix in (b) can occur in the process of recovering $\mathbf{x}$.

Proof. (Stewart) First, we observe that statements (a) and (b) imply each other. For example, if we could prove (a), then we would observe that

$$\|A(A^T D^{-1} A)^{-1} A^T D^{-1}\| \le \|A\| \cdot \|(A^T D^{-1} A)^{-1} A^T D^{-1}\|,$$

and thus we could take $\bar{\chi}_A$ to be $\|A\| \cdot \chi_A$. Similarly, the following inequality allows us to derive (a) from (b):

$$\|(A^T D^{-1} A)^{-1} A^T D^{-1}\| \le \|(A^T A)^{-1} A^T\| \cdot \|A(A^T D^{-1} A)^{-1} A^T D^{-1}\|.$$

Thus, we will prove (b) only. We first need a preliminary lemma, which is also due to Stewart. This lemma is cited later in the paper.

Lemma 1 (Stewart). *Define two subsets X and Y of $\mathbb{R}^m$ as follows:*

$$X = \{\mathbf{z} : \mathbf{z} = A\mathbf{w} \text{ for some } \mathbf{w}, \text{ and } \|\mathbf{z}\| = 1\}$$

and

$$Y = \{\mathbf{z} : A^T D^{-1}\mathbf{z} = \mathbf{0} \text{ for some } D \in \mathcal{D}\}.$$

Then $X \cap \bar{Y} = \emptyset$.

Here, $\bar{Y}$ denotes the closure of Y in the topological sense.

Proof. (Stewart) Suppose there were a $\mathbf{z} \in X \cap \bar{Y}$, so that $\mathbf{z} = A\mathbf{w}$ for some $\mathbf{w}$ and $\|\mathbf{z}\| = 1$. The statement that $\mathbf{z}$ is in $\bar{Y}$ means that there exists an infinite sequence of vectors $\mathbf{z}_1, \mathbf{z}_2, \ldots$ converging to $\mathbf{z}$, and an infinite sequence of matrices $D_1, D_2, \ldots$ such that $A^T D_k^{-1} \mathbf{z}_k = \mathbf{0}$ for all k. Taking the inner product of this equation with $\mathbf{w}$, and substituting $\mathbf{w}^T A^T = \mathbf{z}^T$, yields $\mathbf{z}^T D_k^{-1} \mathbf{z}_k = 0$ for all k. But since $\mathbf{z}_k$ converges to $\mathbf{z}$, there is some k large enough such that for all nonzero coordinate entries of $\mathbf{z}$, the corresponding entry of $\mathbf{z}_k$ has the same sign. This means that $\mathbf{z}^T D_k^{-1} \mathbf{z}_k > 0$, contradicting the previous equation. ∎

Now we use the lemma to conclude the proof of Theorem 1. Since X is compact and $\bar{Y}$ is closed, and since they are disjoint, there is a positive lower bound, say ρ, on the distance between these spaces measured in the $\|\cdot\|$ norm.

Next, choose some $D \in \mathcal{D}$. Choose an arbitrary vector $\mathbf{x}$. Let

$$\mathbf{w} = (A^T D^{-1} A)^{-1} A^T D^{-1} \mathbf{x}.$$

Let $\mathbf{y} = A\mathbf{w}$. The goal is to get an upper bound on $\mathbf{y}$ in terms of $\mathbf{x}$. A one-line calculation shows that $A^T D^{-1}(\mathbf{x} - \mathbf{y})$ is zero. Thus, let $\mathbf{v} = \mathbf{x} - \mathbf{y}$, so that $A^T D^{-1} \mathbf{v} = \mathbf{0}$. Set $t = 1/\|\mathbf{y}\|$. Then we have

$$t\mathbf{v} + t\mathbf{y} = t\mathbf{x}.$$

Note that $t\mathbf{v}$ is in Y and $-t\mathbf{y}$ is in X. Thus, the norm of the left-hand side of this equation is at least ρ. Thus, $\|t\mathbf{x}\| \geq \rho$, i.e.

$$\rho \|\mathbf{y}\| \leq \|\mathbf{x}\|.$$

This gives us the upper bound of $1/\rho$ on $\bar{\chi}_A$, and concludes the proof of the theorem. ∎

Stewart gives an estimate for $\bar{\chi}_A$ which was later proved by O'Leary [1990] to be an exact formula. The formula seems to requires an exponential number of steps to compute $\bar{\chi}_A$. Moreover, this formula is not completely constructive in finite precision arithmetic because of the observation due to Stewart that the parameters χ_A and $\bar{\chi}_A$ in Theorem 1 do not depend continuously on A. In particular, Stewart shows that $\bar{\chi}_A$ for $A = [0, 1]^T$ is 1, but $\bar{\chi}_A$ for $A = [\epsilon, 1]^T$ tends to infinity as ϵ tends to zero.

It would be very interesting if there were an algorithm to compute or approximate $\chi_A, \bar{\chi}_A$ overcoming either of these difficulties (i.e., the algorithm gives the right answer in finite-precision arithmetic, or the algorithm requires only a polynomial number of steps, or both). In Section 7, we give a bound for $\chi_A, \bar{\chi}_A$ for a special class of matrices.

One would like to apply Theorem 1 to give algorithm stability results for finite-precision arithmetic. Here is an example of a straightforward result in this direction. For this theorem, $\|\cdot\|$ is restricted to being a p-norm. (In a p-norm we know the norm of a diagonal matrix explicitly.)

Theorem 2. *Let* $\mathbf{y}$ *be the second component of the exact solution to* (1) *(when* $\mathbf{c} = \mathbf{0}$*). Let* $\hat{\mathbf{y}}$ *be a computed solution to this system, such that* $\hat{\mathbf{y}}$ *is the exact solution to the perturbed system*

$$\begin{pmatrix} D+E & -A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{y}} \end{pmatrix} = \begin{pmatrix} \mathbf{b}+\mathbf{e} \\ \mathbf{0} \end{pmatrix} \tag{4}$$

where E *satisfies the bound* $|E| \le \epsilon \cdot |D|$ *and* $\|\mathbf{e}\| \le \epsilon \|\mathbf{b}\|$, *where* $\epsilon < 1/3$. *Then*

$$\|\mathbf{y} - \hat{\mathbf{y}}\| \le \chi_A \epsilon \cdot \|\mathbf{b}\| \cdot (\bar{\chi}_A + 8/3). \tag{5}$$

The strength of this theorem is that the error bound in (5) is independent of the condition number of D.

Note that we have used the matrix absolute value notation from Golub and Van Loan [1989]. Thus, the assumption $|E| \le \epsilon \cdot |D|$ implies that E is also diagonal.

Proof. By elimination as described above, we have the following formulas for $\mathbf{y}$ and $\hat{\mathbf{y}}$:

$$A^T D^{-1} A\mathbf{y} = -A^T D^{-1}\mathbf{b}$$

and

$$A^T D^{-1} A\hat{\mathbf{y}} + A^T F A\hat{\mathbf{y}} = -A^T D^{-1}(\mathbf{b}+\mathbf{e}) - A^T F(\mathbf{b}+\mathbf{e}).$$

Here, F denotes $(D+E)^{-1} - D^{-1}$, i.e., $ED^{-1}(D+E)^{-1}$. Subtracting these equations yields

$$A^T D^{-1} A(\hat{\mathbf{y}} - \mathbf{y}) + A^T F A\hat{\mathbf{y}} = -A^T D^{-1}\mathbf{e} - A^T F(\mathbf{b}+\mathbf{e}).$$

We can add and subtract $A^T F A\mathbf{y}$ on the left to obtain

$$A^T D^{-1} A(\hat{\mathbf{y}} - \mathbf{y}) + A^T F A(\hat{\mathbf{y}} - \mathbf{y}) + A^T F A\mathbf{y} = -A^T D^{-1}\mathbf{e} - A^T F(\mathbf{b} + \mathbf{e}),$$

which can be written

$$A^T(D^{-1} + F)A(\hat{\mathbf{y}} - \mathbf{y}) = -A^T F A\mathbf{y} - A^T D^{-1}\mathbf{e} - A^T F(\mathbf{b} + \mathbf{e}).$$

Define $G = D^{-1} + F = (D + E)^{-1}$, another diagonal matrix. Introduce new vectors $\mathbf{e}_1 = G^{-1} F A\mathbf{y}$, $\mathbf{e}_2 = G^{-1} D^{-1}\mathbf{e}$, and $\mathbf{e}_3 = G^{-1} F(\mathbf{b} + \mathbf{e})$. Then we can rewrite the preceding equation as

$$A^T G A(\hat{\mathbf{y}} - \mathbf{y}) = -A^T G(\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3).$$

Now we can conclude from Theorem 1 that

$$\|\hat{\mathbf{y}} - \mathbf{y}\| \leq \chi_A(\|\mathbf{e}_1\| + \|\mathbf{e}_2\| + \|\mathbf{e}_3\|).$$

Thus, the theorem is proved once we analyze the three terms on the right-hand side. First, we have

$$\begin{aligned}
\|\mathbf{e}_1\| &= \|(D + E)F A\mathbf{y}\| \\
&= \|(D + E)^{-1} E A\mathbf{y}\| \\
&\leq \|E D^{-1}\| \cdot \|A\mathbf{y}\| \\
&\leq \epsilon\|A\mathbf{y}\| \\
&\leq \epsilon \cdot \bar{\chi}_A \cdot \|\mathbf{b}\|.
\end{aligned}$$

Next, we have

$$\begin{aligned}
\|\mathbf{e}_2\| &= \|G^{-1} D^{-1}\mathbf{e}\| \\
&= \|(D + E)D^{-1}\mathbf{e}\| \\
&\leq \|(D + E)D^{-1}\| \cdot \|\mathbf{e}\| \\
&\leq (1 + \epsilon) \cdot \epsilon \cdot \|\mathbf{b}\|.
\end{aligned}$$

Note that $1 + \epsilon \leq 4/3$. Finally,

$$\begin{aligned}
\|\mathbf{e}_3\| &= \|G^{-1} F(\mathbf{b} + \mathbf{e})\| \\
&\leq \epsilon \cdot (1 + \epsilon) \cdot \|\mathbf{b}\|.
\end{aligned}$$

As above, $1 + \epsilon \leq 4/3$. This proves the theorem. ∎

For the rest of the paper, we regard (5) as the "ideal" bound that could be satisfied by a finite-precision algorithm. The stability result of this theorem is somewhat different from well-known perturbation results. Usually, a bound on the *relative* error in $\mathbf{y}$ is obtained in terms of ϵ and the other parameters. A relative error bound on $\mathbf{y}$ is not possible for the equation at hand. Such a bound would mean that it is not possible for $\|\mathbf{y}-\hat{\mathbf{y}}\|$ to be large when $\|\mathbf{y}\|$ is small. However, one can select a nonzero $\mathbf{b}$ to make $\mathbf{y}$ arbitrarily small—zero in fact—because $\mathbf{b}$ in general has more dimensions than $\mathbf{y}$. Thus, the bound in the theorem seems to be the best form we could hope for.

We will say that an algorithm for (1) is **stable** if, in the presence of finite-precision arithmetic, an error bound of the same form as (5) is satisfied, where ϵ is on the order of the machine roundoff. In particular, the error bound should have the form

$$\|\mathbf{y}-\hat{\mathbf{y}}\| \leq \epsilon \cdot f(A) \cdot \|\mathbf{b}\|,$$

where $f(A)$ is some function of A not depending on D.

If we could show that a finite-precision algorithm satisfied the hypothesis of the theorem (i.e., the computed solution satisfied (4)), then it would automatically be stable. Unfortunately, we do not know of any finite-precision algorithm to solve (1) that achieves the conditions of this theorem. In particular, it seems that the error bound for any obvious algorithm would involve a backwards error term affecting not only D but also A. Since χ_A is not continuous with respect to changes in A, the theorem cannot be extended to the case when A is also perturbed. In Section 5 we derive other conditions to yield an algorithm that is stable.

3 Further development of the theory

In this section, we develop the theory further before turning to algorithms in the next section. First, we make a few remarks about the difference between χ_A and $\bar{\chi}_A$. Then we look at the relationship between $\chi_A, \bar{\chi}_A$ and the corresponding variables for the *dual* problem.

The two parameters $\chi_A, \bar{\chi}_A$ have different properties with respect to changes in A. If A is multiplied by a constant, then χ_A scales in a reciprocal manner. Parameter $\bar{\chi}_A$ is unchanged when A is scaled. It should be apparent from (5) that the ideal situation is when both constants are small. Thus, we will think of A as "well-conditioned" if $\bar{\chi}_A$ is on the order of unity, and χ_A is on the order of $1/\|A\|$. Note that $\bar{\chi}_A$ is at least 1 in any norm: this is seen by taking D to be the identity matrix in Theorem 1(b), in which case $A(A^TA)^{-1}A^T$ is an orthogonal projection matrix with spectral radius of 1.

In the last paragraph we observed that $\bar{\chi}_A$ is unchanged by scaling. More generally, one sees by inspection that $\bar{\chi}_A$ remains unchanged if A is replaced by AR where R is any nonsingular $n \times n$ matrix. This is the same as saying that $\bar{\chi}_A$ depends only on the span of the columns of A, and not the particular basis selected for that span. The parameter χ_A, on the other hand, depends also on the degree of linear independence of the columns of A.

In the proof in the preceding section, we bounded $\bar{\chi}_A$ in terms of $1/\rho$. We now extend that result to show that this bound is an equation: the supremum of $\|A(A^TD^{-1}A)^{-1}A^TD^{-1}\|$ taken over $D \in \mathcal{D}$ is exactly $1/\rho$. We already proved that this supremum is at most $1/\rho$. For the other direction, we observe that if $\mathbf{x} \in X$ and $\mathbf{y} \in Y$, where X, Y are as in Lemma 1, then the following argument puts a lower bound on $\mathbf{x} - \mathbf{y}$. Let D be chosen so that $A^TD^{-1}\mathbf{y} = \mathbf{0}$, and let $\mathbf{w}$ be chosen so that $\mathbf{x} = A\mathbf{w}$. Then

$$A^TD^{-1}(\mathbf{x} - \mathbf{y}) = A^TD^{-1}\mathbf{x} = A^TD^{-1}A\mathbf{w}$$

so

$$\mathbf{w} = (A^TD^{-1}A)^{-1}A^TD^{-1}(\mathbf{x} - \mathbf{y}).$$

This yields

$$\mathbf{x} = A\mathbf{w} = A(A^TD^{-1}A)^{-1}A^TD^{-1}(\mathbf{x} - \mathbf{y}),$$

so

$$\|\mathbf{x}\| \le \bar{\chi}_A \cdot \|\mathbf{x} - \mathbf{y}\|.$$

But $\|\mathbf{x}\| = 1$ by the assumption that $\mathbf{x} \in X$, so we see that $1/\bar{\chi}_A$ is a lower bound on $\|\mathbf{x} - \mathbf{y}\|$.

We now look into the relationship between (1) and its dual problem. Let Z be a *nullspace basis* for A^T, that is, an $m \times (m - n)$ matrix Z of full column rank such that $A^TZ = 0$. We want to know the relationship between $\bar{\chi}_A$ and $\bar{\chi}_Z$. Note that A is a nullspace basis for Z^T, so any relationship between $\bar{\chi}_A, \bar{\chi}_Z$ will also hold if A, Z are interchanged. Note also that $\bar{\chi}_Z$ depends only on the span of the columns of Z: in other words, it does not matter which nullspace basis Z is selected.

In order to obtain a bound for $\bar{\chi}_Z$, we consider two algorithms for solving the following version of (1), where $\mathbf{x}_0$ is some arbitrary m-vector.

$$\begin{pmatrix} D & -A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ A^T\mathbf{x}_0 \end{pmatrix} \tag{6}$$

First, observe that the second equation of (6) is $A^T\mathbf{x} = A^T\mathbf{x}_0$, i.e., $A^T(\mathbf{x} - \mathbf{x}_0) = \mathbf{0}$. This is the same as saying that $\mathbf{x} - \mathbf{x}_0$ is in the nullspace of A, i.e., there is an $m - n$-vector $\mathbf{q}$ such that $\mathbf{x} = \mathbf{x}_0 + Z\mathbf{q}$. Substituting this into the first equation yields $D(\mathbf{x}_0 + Z\mathbf{q}) - A\mathbf{y} = \mathbf{0}$. Multiply by Z^T, using the fact that $Z^TA = 0$, to obtain $Z^TDZ\mathbf{q} = -Z^TD\mathbf{x}_0$. Solving for $\mathbf{q}$ yields $\mathbf{q} = -(Z^TDZ)^{-1}Z^TD\mathbf{x}_0$. Substituting this formula for $\mathbf{q}$ into $\mathbf{x} = \mathbf{x}_0 + Z\mathbf{q}$ yields,

$$\mathbf{x} = -Z(Z^TDZ)^{-1}Z^TD\mathbf{x}_0 + \mathbf{x}_0.$$

Since $\mathbf{x}_0$ is arbitrary, we see that:

$$\|Z(Z^TDZ)^{-1}Z^TD\| = \sup\left\{\frac{\|\mathbf{x} - \mathbf{x}_0\|}{\|\mathbf{x}_0\|} : \mathbf{x}_0 \neq \mathbf{0}; \mathbf{x} \text{ solves } (6)\right\}.$$

This means that

$$\bar{\chi}_Z = \sup\left\{\frac{\|\mathbf{x} - \mathbf{x}_0\|}{\|\mathbf{x}_0\|} : D \in \mathcal{D}; \mathbf{x}_0 \neq \mathbf{0}; \mathbf{x} \text{ solves } (6)\right\}.$$

By applying the triangle inequality, we obtain the bound

$$\bar{\chi}_Z \leq \sup\left\{1 + \frac{\|\mathbf{x}\|}{\|\mathbf{x}_0\|} : D \in \mathcal{D}; \mathbf{x}_0 \neq \mathbf{0}; \mathbf{x} \text{ solves } (6)\right\}. \tag{7}$$

Now, we compute this supremum by solving (6) in a different way. The first equation of (6) is $D\mathbf{x} - A\mathbf{y} = \mathbf{0}$. Multiplying by A^TD^{-1} yields $A^T\mathbf{x} - A^TD^{-1}A\mathbf{y} = \mathbf{0}$. Substituting the second equation yields $A^T\mathbf{x}_0 - A^TD^{-1}A\mathbf{y} = \mathbf{0}$. Solving for $\mathbf{y}$ yields $\mathbf{y} = (A^TD^{-1}A)^{-1}A^T\mathbf{x}_0$. Use the fact that $\mathbf{x} = D^{-1}A\mathbf{y}$ to obtain $\mathbf{x} = D^{-1}A(A^TD^{-1}A)^{-1}A^T\mathbf{x}_0$. Thus,

$$\frac{\|\mathbf{x}\|}{\|\mathbf{x}_0\|} \leq \|D^{-1}A(A^TD^{-1}A)^{-1}A^T\|. \tag{8}$$

Observe that the matrix whose norm we are taking on the right-hand side of (8) is precisely the transpose of the matrix used to define $\bar{\chi}_A$ in Theorem 1(b).

Thus, we see that the norm on the right-hand side is bounded by $\gamma_m \bar{\chi}_A$, where γ_m is the multiplicative factor that makes the following inequality a true statement for all matrices B:

$$\|B^T\| \leq \gamma_m \|B\|.$$

For example, $\gamma_m = 1$ if we use the matrix 2-norm, and $\gamma_m = m$ if we use the 1-norm or ∞-norm. Substitute (8) into (7) to obtain:

$$\bar{\chi}_Z \leq 1 + \gamma_m \bar{\chi}_A.$$

This gives a relationship between $\bar{\chi}_Z$ and $\bar{\chi}_A$. A formula for χ_Z in terms of $\bar{\chi}_A$ is not possible without making further assumptions (for example, that $Z^T Z$ is well-conditioned).

4 The standard algorithms

In this section we describe three standard algorithms for (1). All three of these algorithms are shown to unstable in the sense of stability given in Section 2. By "standard" we mean that these algorithms are described in optimization textbooks such as Coleman [1984], Gill, Murray and Wright [1981], and Fletcher [1987], electrical engineering textbooks such as Chua, Desoer, and Kuh [1987], or civil engineering textbooks such as Timoshenko and Young [1965].

4.1 Symmetric indefinite factorization

Observe in (1) that if we replace $\mathbf{y}$ by $-\mathbf{y}$, the coefficient matrix becomes the symmetric matrix

$$\begin{pmatrix} D & A \\ A^T & 0 \end{pmatrix}.$$

Therefore, we could think about solving this equation with a standard "stable" algorithm for symmetric indefinite linear systems. (See Golub and Van Loan [1989] for some algorithms for this purpose.) Symmetric factorization has the disadvantage that it does not respect the block of zeros, and therefore requires more operations than necessary. In the electrical engineering context, solving (1) directly is known as *sparse tableau analysis.* In optimization, this algorithm is called the *augmented system* method.

This algorithm should be rejected for the following numerical reason: The algorithm "mixes" $\mathbf{b}, \mathbf{c}$ on the right and $\mathbf{x}, \mathbf{y}$ on the left during the forward substitution and back-substitution process. This does not make sense physically, because $\mathbf{b}, \mathbf{c}$ and $\mathbf{x}, \mathbf{y}$ have different physical units and could be on vastly different scales numerically.

To put it another way, this algorithm is sensitive to scaling matrix D by a constant multiple. On the other hand, $\mathbf{y}$ is mathematically unchanged by such a scaling. Indeed, for a simple electrical circuit depicted in Fig. 1, we scaled D by varying amounts and applied the Parlett-Reid [1970] symmetric factorization. Substantially different values of $\mathbf{y}$ (the voltages) were recovered, some accurate to less than one decimal place.

These tests, as well as the others in this paper, were conducted in MATLAB. MATLAB, a software package for interactive numerical computation, is a trademark of The Mathworks, Inc. All the computations were done in IEEE double-precision arithmetic, with about 15 decimal digits of accuracy.

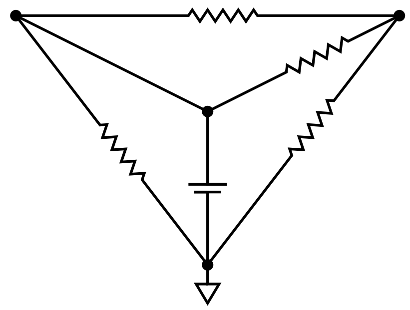
Standard stability results for algorithms like the Parlett-Reid algorithm do not apply, because of our stated interest to recover $\mathbf{y}$ accurately, which is only a part of the solution vector. Moreover, the accuracy of these algorithms depends on the condition number of the equilibrium system (1), which is linked to the condition number of D.

For the problem in this example, $\chi_A, \bar{\chi}_A$ are both small because A is a reduced node-arc incidence matrix (see Section 7). Thus, the failure of this algorithm to satisfy something like (5) indicates that the algorithm is unstable, not that the problem is poorly posed.

Björck [1992] has analyzed symmetric factorization of (1); his work attempts to identify the proper selection of the scaling factor to be applied to D in order to obtain $\mathbf{x}$ or $\mathbf{y}$ accurately. Even with Björck's choice of scaling factor, however, we do not see any evidence that the resulting algorithm will be stable in the sense proposed in Section 2.

4.2 The range-space method

This method obtains $\mathbf{y}$ by solving (2) explicitly. Since $A^T D^{-1} A$ is positive definite, (2) can be solved with Cholesky factorization. Note that this method, as well as the nullspace method described below, is insensitive to scaling of D by a constant multiple.

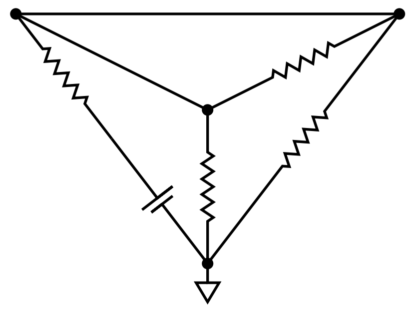


(a)

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}, D = \begin{pmatrix} 10^{-15} & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 10^{-15} & & \\ & & & & 1 & \\ & & & & & 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

(b)

Figure 1: The circuit depicted in (a) yields an equilibrium system of the form (1) with $D, A, \mathbf{b}$, as in (b). Here, all the resistors have size 1, and wires without an indicated resistor have resistance 10^{-15}. The correct values of the voltages of the three nodes are $(1.00, 1.00, 0.67)$ accurate to two digits. The Parlett-Reid algorithm, however, yields $(1.31, 1.31, 0.88)$ when D is replaced by $10^{20}D$, and yields $(0.86, 0.98, 0.58)$ when D is replaced by $10^{25}D$.



(a)

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}, D = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 10^{-15} & & \\ & & & & 10^{-15} & \\ & & & & & 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

(b)

Figure 2: The circuit depicted in (a) yields an equilibrium system of the form (1) with $D, A, \mathbf{b}$, as in (b). Here, all the resistors have size 1, and wires without an indicated resistor have resistance 10^{-15}. The correct values of the voltages of the three nodes are $(0.33, 0.33, 0.33)$ accurate to two digits. The range-space algorithm, however, yields $(0.28, 0.28, 0.28)$.

This method is called the *range-space* method in optimization. In structural analysis it is known as the *displacement* method. In electrical engineering it is called the *nodal analysis* method. In finite elements, the matrix $A^T D^{-1} A$ is known as the *assembled stiffness matrix.* This matrix is known as the *Schur complement* in optimization.

It should be apparent that if D is severely ill-conditioned, then $A^T D^{-1} A$ can also be severely ill-conditioned. This means that inaccurate answers (i.e., answers not respecting (5)) may be obtained. An example of a circuit solution obtained with the range-space method is indicated in Fig. 2. Observe that not even one significant decimal place is obtained in the answer.

In our testing of the range-space method, we observed that the choice of numbering for edges and nodes affected the resulting answer, and some numberings for the example in Fig. 2 gave better answers than the numbering depicted. It is known in the optimization community (Y. Li, private communication) that a good ordering in optimization problems is the one in which the elements of D are sorted with the largest first. This is related to orderings for least-squares problems (see Van Loan [1985], and Powell and Reid's [1969] analysis of Businger and Golub's [1965] algorithm). Nonetheless, even with a different ordering, we do not see any evidence that the resulting algorithm will be stable in the sense of Section 2, because $A^T D^{-1} A$ is ill-conditioned regardless of the ordering.

4.3 The nullspace method

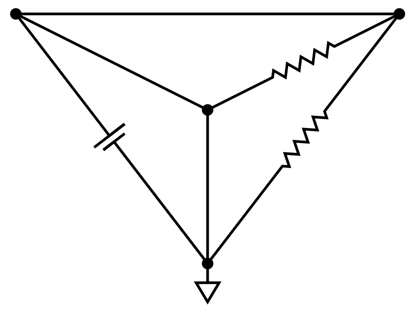
This method obtains $\mathbf{x}$ first, and then $\mathbf{y}$. The essential ingredients of the nullspace method were already described in Section 3. Following Fletcher, we can describe this method as follows. First, a pair of matrices Y, Z is obtained, where Y is an $m \times n$ matrix such that $A^T Y = I$, the identity, and Z is a nullspace basis, that is, an $m \times (m-n)$ of full column rank such that $A^T Z = 0$. Matrix Y is called a *right-inverse* of A^T.

The most stable way to obtain Y, Z is via a QR factorization of A, say $A = QR$. Then Z is set to the last $m-n$ columns of Q, and Y is set to the product $Q_1 R_1^{-T}$, where Q_1 is the first n columns of Q, and R_1 is the first n rows of R.

Once Y, Z are obtained, we observe that the equation $A^T \mathbf{x} = \mathbf{0}$ means $\mathbf{x} = Z\mathbf{q}$ for some $(m-n)$-vector $\mathbf{q}$. This is because Z spans the nullspace of A. Substituting $\mathbf{x} = Z\mathbf{q}$ into the equation $D\mathbf{x} - A\mathbf{y} = \mathbf{b}$ yields $DZ\mathbf{q} - A\mathbf{y} = \mathbf{b}$. Now multiply by Z^T, using the fact that $Z^T A = 0$, to obtain $Z^T DZ\mathbf{q} = Z^T \mathbf{b}$. The matrix $Z^T DZ$ is known as the *reduced Hessian* in the optimization literature.

The linear system $Z^T DZ\mathbf{q} = Z^T \mathbf{b}$ is symmetric and positive definite, and hence may be solved with Cholesky factorization to obtain $\mathbf{q}$. Once $\mathbf{q}$ is obtained, $\mathbf{x}$ can be obtained with the formula $\mathbf{x} = Z\mathbf{q}$. Finally, $\mathbf{y}$ is obtained from the equation $D\mathbf{x} - A\mathbf{y} = \mathbf{b}$ by multiplying through by Y^T, yielding $\mathbf{y} = Y^T(D\mathbf{x} - \mathbf{b})$.

This method is known as the *nullspace method* in optimization. It is called the *force method* in the civil engineering, and *loop analysis* in the electrical



(a)

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}, D = \begin{pmatrix} 10^{-16} & & & & & \\ & 1 & & & & \\ & & 10^{-16} & & & \\ & & & 10^{-16} & & \\ & & & & 10^{-16} & \\ & & & & & 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

(b)

Figure 3: The circuit depicted in (a) yields an equilibrium system of the form (1) with $D, A, \mathbf{b}$, as in (b). Here, all the resistors have size 1, and wires without an indicated resistor have resistance 10^{-16}. The correct values of the voltages of the three nodes are $(0.67, 0.33, 0.67)$ accurate to two digits. The nullspace algorithm, however, yields $(0.89, 0.52, 1.16)$.

engineering.

It should be apparent that this method suffers from the same flaw as the range-space method. The matrix Z^TDZ could be arbitrarily ill-conditioned, and hence solving a linear system with this matrix could give a highly inaccurate answer. Indeed, for the example in Fig. 3, the nullspace method returned an answer for $\mathbf{y}$ without any digits of accuracy.

5 A stable hybrid algorithm

We now describe a stable algorithm for (1). We call this algorithm "hybrid" because it works with both the range-space and nullspace of A. The scaling matrix D is applied to the nullspace. Thus, the particular method described in this section is referred to as the "nullspace-scaled hybrid" (NSH) method, to distinguish it from other hybrid methods that we introduce later.

The first step of the NSH method is to compute a nullspace basis V for $A^T D^{-1}$. Thus, V is an $m \times (m-n)$ matrix of rank $m-n$ such that $A^T D^{-1} V = 0$. The matrix V should be multiplied by a constant so that $\|A\| = \|V\|$.

Then the linear system

$$[A, V] \begin{pmatrix} \mathbf{y} \\ \mathbf{q} \end{pmatrix} = -\mathbf{b} \tag{9}$$

is solved for $\mathbf{y}$ and $\mathbf{q}$. We claim that $\mathbf{y}$ is indeed a solution for the original problem. This linear system can be rewritten as $A\mathbf{y} + V\mathbf{q} = -\mathbf{b}$. Multiplying by $A^T D^{-1}$ causes the second term on the left to drop out, yielding $A^T D^{-1} A\mathbf{y} = -A^T D^{-1}\mathbf{b}$. Thus, this method solves (2) without explicitly forming $A^T D^{-1} A$. Notice that D does not appear at all in the linear system (9).

The NSH method does not appear in standard textbooks, but it has appeared in the literature. For example, Coleman and Li [1989] suggest a similar approach (called the "full-space" method) for optimization, but with the scaling done in a different manner.

We now investigate the numerical stability of this method. We start with a preliminary lemma. For the rest of this section $\|\cdot\|$ is some p-norm.

Lemma 2. *Assume the null basis V computed above is normalized so that $\|V\| = \|A\|$. Suppose also that V is well-conditioned, in the sense that there is a constant $\theta > 0$ such that $\|V\mathbf{x}\| \geq (\|V\| \cdot \|\mathbf{x}\|)/\theta$ for all $\mathbf{x}$. Let $M = [A, V]$. Then*

$$\kappa(M) \leq 2\chi_A \cdot \|A\| + 2\theta \cdot (1 + \bar{\chi}_A).$$

We denote the right-hand side of this inequality by κ_A.

Proof. We use the notation $\kappa(M)$ to denote the condition number of M, that is, $\|M\| \cdot \|M^{-1}\|$. The basic idea of this proof is that the condition number of

$M = [A, V]$ depends on three things: the condition of A, the condition of V (i.e., the value of θ), and the angle between the span of A and the span of V. But this angle cannot be too small because of Stewart's lemma in Section 2.

Suppose that $M\mathbf{z} = \mathbf{c}$ for an arbitrary $\mathbf{c}$. Split $\mathbf{z}$ into two components, say $\mathbf{y}$ and $\mathbf{q}$. Then $A\mathbf{y} + V\mathbf{q} = \mathbf{c}$. Multiply by $A^T D^{-1}$ to obtain $A^T D^{-1} A\mathbf{y} = A^T D^{-1}\mathbf{c}$. Thus, $\|\mathbf{y}\| \le \chi_A \|\mathbf{c}\|$ and $\|A\mathbf{y}\| \le \bar{\chi}_A \|\mathbf{c}\|$ by Stewart's theorem. Next, $V\mathbf{q} = \mathbf{c} - A\mathbf{y}$, so

$$\|V\mathbf{q}\| \le \|\mathbf{c}\| + \bar{\chi}_A \|\mathbf{c}\|.$$

Finally,

$$\begin{aligned} \|\mathbf{q}\| &\le \frac{\theta}{\|V\|} \cdot \|V\mathbf{q}\| \\ &\le \frac{\theta}{\|A\|} \cdot (1 + \bar{\chi}_A) \cdot \|\mathbf{c}\|. \end{aligned}$$

Thus, we have

$$\begin{aligned} \|\mathbf{z}\| &\le \|\mathbf{y}\| + \|\mathbf{q}\| \\ &\le \|\mathbf{c}\| \cdot \left(\chi_A + \frac{\theta}{\|A\|} \cdot (1 + \bar{\chi}_A) \right). \end{aligned}$$

Since $\|M^{-1}\|$ is the supremum of $\|\mathbf{z}\|$ in the case that $\|\mathbf{c}\| = 1$, the parenthesized factor on the left hand side is an upper bound on $\|M^{-1}\|$.

Now finally, observe that $\|M\| \le 2\|A\|$. Thus, multiplying through by $2\|A\|$ gives the result. ∎

Note that the condition that $\|A\| = \|V\|$ does not have to be satisfied exactly; it suffices in the above argument to have A and V on the same order.

The scalar κ_A in the preceding lemma has a complicated form. One sees, however, that it is independent of D, and is invariant if A is multiplied by a constant. We now can state the main theorem about the stability of the NSH method.

Theorem 3. *Assume V satisfies the conditions of the previous lemma. Suppose that the algorithm to solve the linear system $A\mathbf{y} + V\mathbf{q} = -\mathbf{b}$ in floating point arithmetic returns approximate solution $(\hat{\mathbf{y}}, \hat{\mathbf{q}})$ such that*

$$(A + E)\hat{\mathbf{y}} + (V + F)\hat{\mathbf{q}} = -\mathbf{b} + \mathbf{e}, \tag{10}$$

where $\|[E, F]\| \leq \epsilon \|[A, V]\|$ *and* $\|\mathbf{e}\| \leq \epsilon \|\mathbf{b}\|$. *Assume that* $\epsilon \leq 1/(3\kappa_A)$. *Then we have the following error bound:*

$$\|\mathbf{y} - \hat{\mathbf{y}}\| \leq 3\epsilon \frac{\kappa_A^2}{\|A\|} \cdot \|\mathbf{b}\| \tag{11}$$

where $\mathbf{y}$ *is the true solution.*

Remarks.

1. Note that (11) indicates that the NSH algorithm is stable, because D does not enter the bound.
2. The theorem seemingly requires that an *exact* nullspace basis V of $A^T D^{-1}$ be computed. We remark further on this in the next section.
3. The condition that the floating-point algorithm satisfies an error bound like (10) is very natural, and any well-known stable algorithm, say Gaussian elimination with partial pivoting, satisfies such a bound.

Proof. Let $\mathbf{z} = (\mathbf{y}^T, \mathbf{q}^T)^T$ and $\hat{\mathbf{z}} = (\hat{\mathbf{y}}^T, \hat{\mathbf{q}}^T)^T$. Following the standard analysis of Golub and Van Loan [1989], Theorem 2.7.2 and using the previous lemma, we obtain the bound:

$$\|\mathbf{z} - \hat{\mathbf{z}}\| \leq 3\epsilon\kappa_A \cdot \|\mathbf{z}\|.$$

But

$$\begin{aligned} \|\mathbf{z}\| &\leq \|M^{-1}\| \cdot \|\mathbf{b}\| \\ &\leq \frac{\kappa_A}{\|A\|} \cdot \|\mathbf{b}\|. \end{aligned}$$

Thus,

$$\|\mathbf{z} - \hat{\mathbf{z}}\| \leq 3\epsilon \frac{\kappa_A^2}{\|A\|} \cdot \|\mathbf{b}\|.$$

Finally, $\|\mathbf{y} - \hat{\mathbf{y}}\|$ is at most $\|\mathbf{z} - \hat{\mathbf{z}}\|$. ∎

6 Obtaining V for Theorem 3

As mentioned in the previous section, Theorem 3 seemingly requires V to be computed exactly. In fact, it is implicit in the theorem that it suffices to compute a matrix $\hat{V}$ such that $\hat{V}+F'$ is a nullspace basis for A^TD^{-1}, where F' is a small error matrix. This is because using $\hat{V}$ instead of V in (9) is equivalent to compounding the error F in (10) with the further small error F', so the same bound holds with a larger constant.

The standard general-purpose algorithms for nullspace bases do not guarantee that the computed nullspace basis is only a small distance from a true nullspace basis (i.e., a forward-error bound). Instead, these algorithms guarantee that the computed nullspace basis is an exact nullspace basis for the perturbed matrix $A^TD^{-1}+E$ (i.e., a backward-error bound). This is not useful in the present context, because adding E to A^TD^{-1} spoils the special structure that allows Stewart's theorem to be applied.

We now describe a technique to obtain a V with a forward error bound. First, we obtain a nullspace basis Z for A^T. Then we obtain a nullspace basis V for A^TD^{-1} with the formula $V=DZR$, where R is another diagonal matrix.

To compute Z, find a subset of rows of n rows of A that form a *basis*, that is, the $n\times n$ submatrix of A determined by those rows is nonsingular. Now, for each of the remaining $m-n$ "nonbasic" rows, solve for the nonbasic row in terms of the basis rows. The coefficients of this dependence form a column vector with m entries, of which at most $n+1$ are nonzero. This yields a *fundamental* nullspace basis Z, that is, a nullspace basis for A^T with an embedded $(m-n)\times(m-n)$ identity matrix.

If we assume that $\bar{\chi}_A$ is reasonably small, then it turns out that this nullspace basis has a special property, namely, all of the nonzero entries have roughly the same magnitude. If we think of a vector $\mathbf{x}$ as being one column of Z, we can state this as a lemma. In this lemma, we assume that the ∞-norm is used throughout. The constant $\bar{\chi}_Z$ arises in this lemma; recall that $\bar{\chi}_Z$ is bounded in terms of $\bar{\chi}_A$, as demonstrated in Section 3.

Lemma 3. *Let* $\mathbf{x}$ *be a nonzero vector such that* $A^T\mathbf{x}=\mathbf{0}$. *Assume that* $\mathbf{x}$ *has nonzero entries in positions* $\{i_1,\ldots,i_p\}$, *where* $\{i_1,\ldots,i_p\}\subset\{1,\ldots,m\}$. *Suppose also that for every subset* I *of size* $p-1$ *of* $\{i_1,\ldots,i_p\}$, *the rows of* A *indexed by* I *are linearly independent. Then the magnitude of the smallest* *nonzero entry in* $\mathbf{x}$ *is no smaller than* $\|\mathbf{x}\|_\infty/\bar{\chi}_Z$.

Proof. Without loss of generality, we can take $\|\mathbf{x}\| = 1$. Let δ be the magnitude of the entry in $\mathbf{x}$ with the smallest nonzero magnitude, and suppose this entry is in position i_p. Since rows $i_1, \ldots, i_{p-1}$ of A are linearly independent, there is a vector $\mathbf{w}$ such that $A\mathbf{w}$ attains arbitrarily specified entries in these $p-1$ positions. In particular, there is a vector $\mathbf{w}$ such that $A\mathbf{w}$ agrees with $\mathbf{x}$ in positions $i_1, \ldots, i_{p-1}$.

Now, let C be a diagonal matrix with entry '1' in positions $i_1, \ldots, i_{p-1}$, and very small entries in the other diagonal positions. Observe that $CA\mathbf{w}$ agrees with $\mathbf{x}$ in positions $i_1, \ldots, i_{p-1}$, and has nearly zeros in all the other positions, including i_p. Let $\mathbf{y} = CA\mathbf{w}$. Then we observe that if we define X, Y as in Lemma 1 for Z, i.e.,

$$X = \{\mathbf{z} : \mathbf{z} = Z\mathbf{w} \text{ for some } \mathbf{w}, \text{ and } \|\mathbf{z}\| = 1\}$$

and

$$Y = \{\mathbf{z} : Z^T C^{-1}\mathbf{z} = \mathbf{0} \text{ for some } C \in \mathcal{D}\}.$$

then $\mathbf{x} \in X$ and $\mathbf{y} \in Y$. This means that $\|\mathbf{x} - \mathbf{y}\|$ is at least $1/\bar{\chi}_Z$. But $\|\mathbf{x} - \mathbf{y}\|$ is arbitrarily close to δ (depending on how close to zero the diagonal entries of C other than $i_1, \ldots, i_{p-1}$ are). Thus, $\delta \geq 1/\bar{\chi}_Z$. This proves the lemma, since we assumed $\|\mathbf{x}\| = 1$. ∎

This lemma has two important consequences. First, the lemma implies that the nullspace basis Z is well conditioned. In particular, with suitable ordering of the rows and columns of Z, we can write it in the form

$$Z = \begin{pmatrix} I \\ W \end{pmatrix} \tag{12}$$

where W is a matrix whose entries are at most $\bar{\chi}_Z$ in magnitude.

Second, the lemma has the following more subtle consequence: if the columns of Z are calculated with a small forward error, this means in fact that these entries have a small *componentwise* forward error. We can express this formally as follows. Let Z be the nullspace basis of A computed with the above algorithm using exact arithmetic, and let $\hat{Z}$ be the nullspace basis computed in the presence of floating point errors. If we assume that the columns of Z are computed with a small forward error, that is,

$$\|\mathbf{x} - \hat{\mathbf{x}}\| \leq \epsilon\|\mathbf{x}\|,$$

where $\mathbf{x}$ is a column of Z and $\hat{\mathbf{x}}$ is a column of $\hat{Z}$, then we immediately obtain the result:

$$|\mathbf{x} - \hat{\mathbf{x}}| \leq \epsilon\bar{\chi}_Z|\mathbf{x}|.$$

The componentwise error bound is important for the following reason: to obtain V, recall that we form the product DZR, where D is the diagonal matrix from (1), and R is a diagonal matrix not yet specified. Because D, R are both diagonal and Z has a small componentwise error, then DZR also will have a small componentwise forward error (and therefore, a small norm forward error) no matter how D, R are chosen.

The argument in the last two paragraphs presupposed that the columns of Z would have a small normwise forward error. A small normwise forward error is obtained if the basis rows of A are well-conditioned, because the columns of Z are solutions to a linear systems involving the basis rows. So, now we ask whether the basis rows of A are well-conditioned. In fact, there is a bound on the condition number of the basis rows. For simplicity of notation, assume the basis rows of A are the last n rows (this corresponds to the ordering in (12)), and let B be the $n \times n$ submatrix of basis rows. Let $\mathbf{b}$ be an arbitrary n-vector, and let $\hat{\mathbf{b}}$ be its extension to $\mathbb{R}^m$ obtained by filling in the first $m - n$ entries with zeros.

Let C be the diagonal matrix with 1's in the last n diagonal positions and very small entries in other positions. Then $A^TCA \approx B^TB$ and $A^TC\hat{\mathbf{b}} \approx B^T\mathbf{b}$, where the approximations can be arbitrarily close as we make diagonal entries of C close to zero. Thus, if vector $\mathbf{x}$ is the solution to $A^TCA\mathbf{x} = A^TC\hat{\mathbf{b}}$, then $B^TB\mathbf{x} \approx B^T\mathbf{b}$. Since B is square and invertible, this last equation means that $B\mathbf{x} \approx \mathbf{b}$. But we also know that $\|\mathbf{x}\| \leq \chi_A\|\hat{\mathbf{b}}\|$, so $\|\mathbf{x}\| \leq \chi_A\|\mathbf{b}\|$. Recall that for A "well-conditioned," we should have $\chi_A = \phi/\|A\|$, where ϕ is a small constant. Finally, $\|A\| \geq \|B\|$ so $\chi_A \leq \phi/\|B\|$. Thus, we have the bound

$$\|\mathbf{x}\| \leq \|\mathbf{b}\| \cdot \phi/\|B\|.$$

Since $B\mathbf{x} \approx \mathbf{b}$ and $\mathbf{b}$ is arbitrary, then we can fix $\|\mathbf{b}\| = 1$ and take the supremum of $\|\mathbf{x}\|$ over all such $\mathbf{b}$'s to obtain the bound:

$$\|B^{-1}\| \leq \phi/\|B\|.$$

Multiplying through by $\|B\|$ shows that B has a bounded condition number.

Thus, we conclude that we can compute Z using the above algorithm such that Z is simultaneously well conditioned and has a small forward componentwise error. Now, we apply D on the left. As we see from (12), the upper $(m-n) \times (m-n)$ block of DZ is some diagonal matrix. Finally, choose a diagonal matrix R so that the upper $(m-n) \times (m-n)$ block of $V = DZR$ is once again the identity matrix. (This uniquely determines R.)

This algorithm gives a nullspace basis V for $A^T D^{-1}$ with a small forward error. Moreover, V has the form

$$V = \begin{pmatrix} I \\ W' \end{pmatrix} \tag{13}$$

There is no reason to believe, however, that V is well conditioned. In fact, W' can have arbitrarily large entries if the basic rows of A have large corresponding entries in D.

We now show how to maintain control over the size of W' by correctly choosing the basis rows in A. (Until now, we have not made any assumptions about the basis rows except that they are independent.) The algorithm for obtaining B is the following "greedy" approach. Assign *weights* to the rows, where the weight of the ith row is the (i,i) entry of D. Now, select the row with the lowest weight and insert it in B. Continue appending the lowest-weighted remaining row of A to B. Before appending a row to B, one checks that the row is linearly independent from the rows already in B. If a dependence is found, then the row is discarded. Continue this process until B has n rows.

Since A is sparse for many applications, this process of testing rows for independence from previous rows should involve both combinatorial and numerical tests for dependence, such as the inner loop of Gilbert and Heath's [1987] algorithm for nullspace bases. Indeed, for RNAI matrices (described in the next section), the process of selecting B is purely combinatorial and very efficient.

We now claim that if B is selected with this greedy approach, then W' in (13) has a small bound. Consider the (i,j) entry of V. Assume $i > m-n$, so that the entry in question falls in the W' portion in (13). Assume that this entry is nonzero. This means that the jth row of A is expressed in terms of rows $m-n+1, \ldots, m$ with a nonzero coefficient for the ith row. We claim that this implies that $d_{ii} \leq d_{jj}$. Suppose not; suppose that $d_{jj} < d_{ii}$. Observe that, by the positions of the rows, i is basic and j is nonbasic. This means that j was passed over by the greedy algorithm for forming the basis, since row i was added to B even though it has a higher weight than row j. This in turn means that row j must have been linearly dependent on rows already in B when it was encountered by the greedy algorithm. But this is impossible, because we already know that row j can be expressed in terms of the basis rows with a nonzero coefficient for row i, so row j cannot be dependent on B until after row i is added to B. Thus, this contradiction shows that $d_{ii} \leq d_{jj}$.

Finally, when we scale Z on the left by D and on the right by R, it is easy to see that the (i, j) entry of V is equal to the (i, j) entry of Z multiplied by d_{ii}/d_{jj}, which we now have shown to be a positive scalar no larger than one. This shows that the entries of W' in (13) are no larger than the corresponding entries of W in (12), and we already have argued that W has a small bound.

If V has the form (13) with a bound on W', then it is well-conditioned in the sense of Theorem 3. Let $\mathbf{x}$ be an arbitrary vector, and consider $\mathbf{y} = V\mathbf{x}$. Observe that $\mathbf{y}$ agrees with $\mathbf{x}$ in its first $m - n$ positions. Thus, $\|\mathbf{y}\| \geq \|\mathbf{x}\|$. On the other hand, $\|V\| \leq 1 + \|W'\|$. Thus, if we take $\theta = 1 + \|W'\|$, then we have the bound

$$\|V\mathbf{x}\| \geq (\|V\| \cdot \|\mathbf{x}\|)/\theta$$

needed for Theorem 3.

The only property remaining that V must have is the equation $\|V\| = \|A\|$. This is easily obtained by scaling V uniformly.

7 RNAI matrices

A special class of matrices A arising in many applications are *reduced node-arc incidence* (RNAI) matrices. Let G be an directed graph, weakly connected, with no parallel edges and no self-loops. (The assumptions about connectivity and parallel edges could be easily removed at the expense of a more complicated exposition.) Assume G has m arcs and $n + 1$ nodes.

The *node-arc incidence* matrix for G is a matrix A_0 with one row for each edge of G and one column for each node. Each row contains two nonzero entries, a '1' and a '-1.' The '-1' entry occurs in the column corresponding to the tail of the arc, and the '1' entry occurs in the column corresponding to its head.

The *reduced node-arc incidence* (RNAI) matrix A for G is obtained by deleting a column of A_0 corresponding to an arbitrary node. The deleted node is called "ground" in an electrical engineering context. In an RNAI matrix, each row has either a '1'/'-1' pair of entries, or a lone '1' or lone '-1' in rows corresponding to arcs with one end grounded.

Electrical networks are the main application in which A of (1) turns out to be an RNAI matrix. Indeed, the matrix A occurring in Fig. 1 to Fig. 3 is an RNAI matrix. Optimization problems with flow constraints can give rise to RNAI or related matrices. Structural analysis in civil engineering can give rise to matrices A that have a block-RNAI structure.

It is a well-known fact of algebraic graph theory (see, for example, Welsh [1976]) that an RNAI matrix A has full column rank. The process of finding a basis among the rows of A corresponds to identifying a spanning tree of G_0, where G_0 is the undirected graph that results when the arcs of G are stripped of their orientation. A *spanning tree* is a subgraph T of G_0 that is incident upon every vertex of G_0, is connected, and has no cycles. The resulting nullspace basis Z has the form (12), with the additional property that every entry in W is either 0, 1, or -1. Thus, there is a very good upper bound on $\|W\|$.

The greedy algorithm described in the previous section corresponds to finding a *minimum-weight* spanning tree. The minimum-weight spanning tree can be computed very efficiently; see, for example, Papadimitriou and Steiglitz [1982]. From this spanning tree we construct Z as described above, and then V by multiplying DZR. The process of obtaining Z and the basis is completely combinatorial for an RNAI matrix.

We have tried this version of the NSH algorithm on all the problems in Section 4 in MATLAB. We used an $O(m \log n)$ algorithm for constructing minimum-weight spanning trees. We obtained answers with about 15 digits of accuracy in all cases.

These results show that we can easily construct a V with the right properties for Theorem 3 for RNAI matrices. A natural followup question is whether the constants in the theorem, namely χ_A and $\bar{\chi}_A$, have reasonable bounds for RNAI matrices. Such bounds are also important because we want to argue that the incorrect answers obtained with the standard methods in Section 4 are due to numerical problems with the algorithms, and not ill-conditioning in A.

In fact, both constants $\chi_A, \bar{\chi}_A$ have small bounds for RNAI matrices, as stated in the following theorem.

Theorem 4. *Let A be an RNAI matrix of size $m \times n$. Then $\chi_A \leq n^2$ and $\bar{\chi}_A \leq n$ if the ∞-norm is used.*

Proof. This proof will directly put a lower bound on the distance between a vector $\boldsymbol{\zeta} \in X$ and $\mathbf{z} \in Y$, where X, Y are as in Lemma 1. This gives a lower bound for ρ in the proof of Theorem 1, and therefore an upper bound for $\bar{\chi}_A$.

We define a *cut* of G to be a partition of its nodes into two disjoint nonempty sets, S and T. The *cut-arcs* are those arcs with one endpoint in S and the other in T. The *forward* cut-arcs are those with tail in S and head in T; the *reverse* cut-arcs are the opposite.

Let $\mathbf{z}$ be a vector such that $A^T\mathbf{z} = \mathbf{0}$, where A is the RNAI matrix of G. Note that $\mathbf{z}$ has m components, and each component of $\mathbf{z}$ is associated with an edge of G. Let (S, T) be an arbitrary cut of G. We claim that the sum of the z_i's associated with forward cut-arcs is exactly equal to the sum of z_i's associated reverse cut-arcs. To see this, assume the ground node is in T, and focus on S (if the ground node were in S, we would focus instead on T). Let $\mathbf{q}$ be the n-vector "indicator" for S that has a 1 in the ith position if the ith node is in S, else a zero. Since $A^T\mathbf{z} = \mathbf{0}$, we have $\mathbf{q}^T A^T \mathbf{z} = \mathbf{0}$. Letting $\mathbf{v}$ denote $A\mathbf{q}$, we have $\mathbf{v}^T\mathbf{z} = \mathbf{0}$.

But there is a combinatorial interpretation for the entries in $\mathbf{v}$. In particular, for an arc with either both endpoints or neither endpoint in S, the corresponding entry of $\mathbf{v}$ is zero. For a cut-arc, the entry of $\mathbf{v}$ is ± 1, with the sign depending on which endpoint is in S. Thus, the statement $\mathbf{v}^T\mathbf{z} = 0$ means precisely that the sum of the $z_i's$ on forward cut-arcs is equal to the sum of z_i's on reverse cut-arcs.

An implication of the result in the last paragraph is as follows. For any cut (S, T), and for any $\mathbf{z}$ such that $A^T\mathbf{z} = \mathbf{0}$, there has to be either a forward cut-arc with a nonnegative value of z_i, or a reverse cut-arc with a nonpositive value of z_i. This implication follows by contradiction: if all the forward cut-arcs have negative z_i's and all the reverse cut-arcs have positive z_i, then their sums could not be equal.

Now, more generally, let $\mathbf{z}$ be a vector such that $A^T D^{-1}\mathbf{z} = \mathbf{0}$, where D is an arbitrary positive definite diagonal matrix. Then, since $\mathbf{z}$ has the same pattern of signs as $D\mathbf{z}$, and $D\mathbf{z}$ is a null vector of A^T, we conclude that for any cut (S, T), $\mathbf{z}$ has the property described in the previous paragraph.

Next, let $\boldsymbol{\zeta}$ be a vector such that $\|\boldsymbol{\zeta}\|_\infty = 1$, and $\boldsymbol{\zeta} = A\mathbf{w}$ for some $\mathbf{w}$. The statement $\boldsymbol{\zeta} = A\mathbf{w}$ has a combinatorial interpretation: each entry of $\mathbf{w}$ may be thought of as "potential" associated with the corresponding node of G (with the "potential" of the ground node being zero). Then if $\boldsymbol{\zeta} = A\mathbf{w}$, we can think of the component of $\boldsymbol{\zeta}$ associated with each arc to be the "potential drop," that is, the difference in potential between the head of the arc and the tail.

Since $\|\boldsymbol{\zeta}\|_\infty = 1$, there is at least one potential drop of magnitude 1. Let us write down the potentials of all the nodes, $w'_1, w'_2, \ldots, w'_{n+1}$, including ground, in decreasing order. In other words, the list $w'_1, w'_2, \ldots, w'_{n+1}$ has the same entries as the vector $(\mathbf{w}, 0)$, but in decreasing sorted order.

Then the difference between the w'_1 and w'_{n+1} is at least 1. This means that, in the list of sorted potentials, there must be a gap somewhere in the list of size at least $1/n$. In other words, there is a j such that $w'_j - w'_{j+1} \geq 1/n$. Now, let (S,T) be the cut such that S contains the nodes associated with $w'_1, \ldots, w'_j$, and T contains the nodes associated with $w'_{j+1}, \ldots, w'_{n+1}$. Then every node in S has potential at least $1/n$ higher than every node in T. This means that the component of $\boldsymbol{\zeta}$ associated with any forward cut-arc is negative with magnitude at least $1/n$, and the component of $\boldsymbol{\zeta}$ associated with any reverse cut-arc is positive with magnitude at least $1/n$.

But finally, this means that if $\mathbf{z}$ is a vector such that $A^T D^{-1}\mathbf{z} = \mathbf{0}$, then for the cut (S,T) constructed in the last paragraph, there is at least one forward cut-arc where the components of $\boldsymbol{\zeta}$ and $\mathbf{z}$ differ by $1/n$, or at least one reverse cut-arc where the components differ by $1/n$.

This shows that if A is an RNAI matrix, then the minimum distance between the two sets X and Y described in Lemma 1 is at least $1/n$ in the ∞-norm. Thus, $\bar{\chi}_A \leq n$,

Now we bound χ_A. The argument so far has shown that if $\mathbf{y}$ is a solution to $A^T D^{-1} A\mathbf{y} = A^T D^{-1}\mathbf{b}$ with $\|\mathbf{b}\|_\infty = 1$, then $\|A\mathbf{y}\|_\infty \leq n$. The remaining task is to get a bound on $\|\mathbf{y}\|_\infty$. Recall the combinatorial interpretation of $\mathbf{y}$ and $A\mathbf{y}$: if $\mathbf{y}$ represents potentials at the nodes, then $A\mathbf{y}$ represents the potential drops. If all potential drops are at most n, the no node could have a potential higher than n^2, since there is a path of at most n arcs between any node and ground. Thus, we see that $\|\mathbf{y}\|_\infty \leq n^2$. ∎

8 Generalizing the problem

In this section we comment on the possibility of lifting some of the assumptions made in Section 1. One assumption made throughout the paper is that we are interested in recovering $\mathbf{y}$ in (1) rather than $\mathbf{x}$. Assuming we have $\mathbf{y}$, $\mathbf{x}$ can be recovered from the equation $D\mathbf{x} - A\mathbf{y} = \mathbf{b}$. It is easy to see that we can compute $D\mathbf{x}$ stably (i.e., with only small normwise error). Then $\mathbf{x}$ is obtained by scaling the computed $D\mathbf{x}$ with D^{-1}. The recovered $\mathbf{x}$ would have a relative error that could be on the order of the condition number of D. This is inherent in the problem—equation (1) is ill-conditioned with respect to $\mathbf{x}$ when D is badly scaled, and there is no theorem like Theorem 1 that holds for $\mathbf{x}$. Thus, we see that recovering $D\mathbf{x}$ accurately is the best that could be hoped for, in the case that $\mathbf{b} \neq \mathbf{0}$, $\mathbf{c} = \mathbf{0}$.

This instability is not merely a numerical artifact—there is a corresponding physical instability. For example, in the context of electrical networks with batteries, it is possible to construct a circuit as follows. One arc of the circuit has very low resistance, but also very small current because the battery voltages are balanced so that the potential across the arc is small. Then a small perturbation to the battery voltages (i.e., $\mathbf{b}$) could cause a great multiplication of the current (i.e., $\mathbf{x}$). On the other hand, $D\mathbf{x}$, the voltage drops across the resistors, would suffer only a small perturbation.

Another assumption was that $\mathbf{c} = \mathbf{0}$. Can we drop this assumption? One special case, which is dual to the case considered for most of this paper, is when $\mathbf{c} \neq \mathbf{0}$, $\mathbf{b} = \mathbf{0}$, and we are interested in recovering $\mathbf{x}$ instead of $\mathbf{y}$. In this case, let Z be a nullspace basis for A^T, and let $\mathbf{x}_0$ be an arbitrary solution to the underdetermined system $A^T\mathbf{x} = \mathbf{c}$. Then the formula for $\mathbf{x}$ is obtained by solving: $Z^T DZ\mathbf{q} = Z^T D\mathbf{x}_0$, and then setting $\mathbf{x} = Z\mathbf{q} + \mathbf{x}_0$. The equation $Z^T DZ\mathbf{q} = Z^T D\mathbf{x}_0$ is of the exact format of (2). Thus, we can apply the machinery developed in this paper, except with Z in place of A, $\mathbf{x}_0$ in place of $\mathbf{b}$, and D in place of D^{-1}. Notice that the nullspace of Z^T is the same as the the range of A. Thus, the version of the hybrid algorithm for this problem would be the range-space scaled hybrid algorithm, RSH. Note also that we have derived relationships between $\bar{\chi}_A$ and $\bar{\chi}_Z$ in Section 3.

In the most general case that $\mathbf{b} \neq \mathbf{0}, \mathbf{c} \neq \mathbf{0}$, it seems unlikely that we could obtain an algorithm that computes either $\mathbf{x}$ or $\mathbf{y}$ accurately when D is ill-conditioned, because both variables become ill-conditioned. A compromise algorithm might be as follows. Let Z be a basis for the nullspace of A^T.

Then solve the linear system:

$$D^{-1/2}A\mathbf{y} + D^{1/2}Z\mathbf{q} = -D^{-1/2}\mathbf{b} + D^{1/2}\mathbf{x}_0$$

where $\mathbf{x}_0$ is as in the previous paragraph. Multiplying through by $A^T D^{-1/2}$ shows that this solves the general case of (1) for $\mathbf{y}$. This method has the advantage that it "splits" the ill-conditioning between the nullspace and the range-space. We could call this the bi-scaled hybrid algorithm, or BSH.

The last restrictive assumption made in Section 1 is that D is diagonal. As mentioned earlier, in finite element and structural problems in two and three dimensions, we can in general assume only that D is block diagonal. Stewart's theorem does not generalize to nondiagonal matrices, as shown in his paper. In the applications of finite elements and structures, however, A has a certain sparsity pattern that is correlated to the blocks of D. Perhaps a result could be established for special structured matrices A.

9 Open questions

Perhaps the number of open questions raised by this work exceeds the number of results we have obtained. Here are some examples of open questions.

1. Is there a good algorithm to obtain $\chi_A, \bar{\chi}_A$? Here "good" means either that the algorithm is polynomial time, or that it gives an approximate answer in finite precision arithmetic, or both.

2. What are useful bounds for $\chi_A, \bar{\chi}_A$ in the case that A has the special structure arising in applications more general than RNAI matrices?

3. Can Stewart's theorem be generalized to a block-diagonal D in the case that the structure of A is correlated to the structure of D?

4. As mentioned in Section 4, there are techniques known in the literature that can be applied to the standard algorithms (like Björck's choice of the scaling factor); these techniques seem to stabilize the standard algorithms. Could it be proved that one of the standard algorithms is stable if one of these techniques is used? The standard algorithms are much simpler than the NSH method.

5. This paper has not dealt at all with issues of algorithmic efficiency, but there are many. For instance, in the NSH method, do we explicitly have to solve the $m \times m$ system (9), or can we approach it implicitly? The algorithm for computing Z described in Section 6 was geared solely towards numerical stability. Recently, Stern and Vavasis [1992] have proposed a way to compute Z for an RNAI matrix based on *separators* of G_0. The Stern-Vavasis algorithm aims for sparsity of Z rather than stability. Is there some combination of the algorithm here with the ideas from the other paper that attains both numerical stability and a sparse nullspace basis Z?

6. For the general case of recovering both $\mathbf{x}$ and $\mathbf{y}$, is the BSH method described in Section 8 any better than the range-space or nullspace methods? We tried the BSH algorithm using a nullspace basis Z generated by QR factorization described in Section 4 (in particular, not any special spanning-tree basis). We obtained very accurate answers for all the problems in which the standard methods failed. Is there an explanation for the success of the BSH method?

7. Are there good iterative methods for (1)? For instance, is there an iterative method whose convergence rate depends only on A, and gives a good solution for $\mathbf{y}$ in the sense of a bound like (5)?

Acknowledgments

The author benefitted from discussion of these ideas with Tom Coleman, Yuying Li, Danny Ralph, Mike Todd, Nick Trefethen and Charlie Van Loan of Cornell, Alan Edelman of Berkeley, Nick Higham of Manchester, and Ilse Ipsen of Yale.

References

A. Björck [1992], Pivoting and stability in the augmented system method, *Numerical Analysis 1991: Proceedings of the 14th Dundee Conference*, Longman Scientific and Technical, Harlow, Essex, England.

P. Businger and G. H. Golub [1965], Linear least squares solutions by Householder matrices, *Numer. Math.* 7:269–276.

S. Chandrasekaran and I. Ipsen [1991], Perturbation theory for the solution of systems of linear equations, RR-866, Department of Computer Science, Yale University, New Haven, CT.

L. O. Chua, C. A. Desoer, and E. S. Kuh [1987], *Linear and Nonlinear Circuits*, McGraw–Hill, New York.

T. F. Coleman [1984], *Large Sparse Numerical Optimization*, Lecture Notes in Computer Science vol. 165, Springer Verlag, Berlin.

T. F. Coleman and Y. Li [1989], A globally and quadratically convergent affine scaling method for linear l_1 problems, Technical Report 89–1026, Department of Computer Science, Cornell University, Ithaca, NY. Also, *Math. Progr.*, to appear.

R. Fletcher [1987], *Practical Methods of Optimization, 2nd Edition*, J. Wiley and Sons, Chichester.

J. Gilbert and M. Heath [1987], Computing a sparse basis for the null space, *SIAM J. Alg. Disc. Meth.* 8:446–459.

P. E. Gill, W. Murray, and M. H. Wright [1981], *Practical Optimization*, Academic Press, London.

G. H. Golub and C. F. Van Loan [1989], *Matrix Computations, 2nd Edition*, Johns Hopkins Univ. Press, Baltimore.

N. Karmarkar [1984], A new polynomial-time algorithm for linear programming, *Combinatorica* 4:373–395.

D. P. O'Leary [1990], On bounds for scaled projects and pseudoinverses, *Linear Algebra and its Applications* 132:115–117.

C. H. Papadimitriou and K. Steiglitz [1982], *Combinatorial Optimization: Algorithms and Complexity*, Prentice–Hall, Englewood Cliffs, N.J.

B. N. Parlett and J. K. Reid [1980], On the solution of a system of linear equations whose matrix is symmetric but not definite, *BIT* 10:386–97.

M. J. D. Powell and J. K. Reid [1969], On applying Householder's method to linear least squares problems, *Proc. IFIP Congress, 1968.*

J. M. Stern and S. A. Vavasis [1992], Nested dissection for sparse nullspace bases, *SIAM J. Matrix Anal. Appl.*, to appear.

G. W. Stewart [1989], On scaled projections and pseudoinverses, *Linear Algebra and its Applications* 112:189–193.

G. Strang [1986], *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, MA.

S. P. Timoshenko and D. H. Young [1965], *Theory of Structures, 2nd Edition*, McGraw–Hill, New York.

M. J. Todd [1990], A Dantzig-Wolfe-like variant of Karmarkar's interior-point linear programming algorithm, *Operations Research* 38:1006–1018.

C. F. Van Loan [1985], On the method of weighting for equality-constrained least-squares problems, *SIAM J. Numer. Anal.* 22:851–864.

D. Welsh [1976], *Matroid Theory,* London Mathematical Society Monograph 8, Academic Press, London.

M. H. Wright [1992], Interior methods for constrained optimization, in *Acta Numerica 1992*, Cambridge University Press, Cambridge.