SIAM J. NUMER. ANAL.
Vol. 29, No. 4, pp. 1166–1186, August 1992

017

A GLOBAL AND QUADRATICALLY CONVERGENT METHOD FOR LINEAR l_∞ PROBLEMS*

THOMAS F. COLEMAN[†] AND YUYING LI[‡]

Abstract. A new globally and quadratically convergent algorithm is proposed for the linear l_∞ problem. This method works on the piecewise linear l_∞ problem directly by generating descent directions—via a sequence of weighted least squares problems—and using a piecewise linear line search to ensure a decrease in the l_∞ function at every step. It is proven that ultimately full Newton-like steps are taken where the Newton step is based on the complementary slackness condition holding at the solution. Numerical results suggest a very promising method; the number of iterations required to achieve high accuracy is relatively insensitive to problem size.

Key words. linear l_∞ estimation, linear programming, interior-point algorithm, simplex method, affine scaling method, discrete Chebyshev problem

AMS(MOS) subject classifications. 65H10, 65K05, 65K10

1. Introduction.

Given a matrix $A \in \Re^{n\times m}$, $m > n$, and a set of data $b \in \Re^m$, a common problem is to find a vector $x \in \Re^n$ such that the linear model $A^T x$ closely matches the given data b. Therefore, the following problem is important:

$$\min_{x\in\Re^n} \|A^T x - b\|$$

where the most often used measures are 2-norm, 1-norm, and ∞-norm. The 2-norm solution, by far the most popular choice, can be obtained in a single step, e.g., using a QR factorization of A^T.

There are situations where it is preferable to use either $\|\cdot\|_1$ or $\|\cdot\|_\infty$; however, the resulting numerical problems are much more difficult. For example, the piecewise linear functions can be minimized by forming an equivalent linear programming problem with special structure. A tailored simplex method can then be used (e.g. [1], [2]). Alternatively, the linear programming formulations can be solved using an interior point method (e.g., [9]).

In both approaches indicated above, the solution techniques are iterative; however, the approaches differ: in the first case the sequence of points generated is finite, whereas in the second, assuming exact real arithmetic, an infinite sequence is generated (theoretically). The sequence produced by an interior point method usually converges linearly: this is one place where an improvement can be made. Indeed, Coleman and Li [5] have developed a globally and quadratically convergent method for l_1 problems. Computational results exhibit quadratic convergence; the method is promising for solving large problems. Recently, Zhang and Tapia [11] also proposed a quadratically convergent primal-dual interior point method for general linear programming; however, no computational results have been reported for that method.

[*] Received by the editors June 1, 1990; accepted for publication (in revised form) July 18, 1991.

[†] Computer Science Department and Center for Applied Mathematics, Cornell University, Ithaca, New York 14853. This research was partially supported by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under grant DE-FG02-86ER25013.A000, by the U.S. Army Research Office through the Mathematical Sciences Institute, Cornell University, and by the Computational Mathematics Program of the National Science Foundation under grant DMS-8706133.

[‡] Computer Science Department, Cornell University, Ithaca, New York 14853. The research of this author was partially supported by the U.S. Army Research Office through the Mathematical Sciences Institute, Cornell University, and by the Computational Mathematics Program of the National Science Foundation under grant DMS-8706133.

In [4] the algorithm of Coleman and Li has been subsequently extended to solve linear programming problems with upper and lower bounds on all variables (A related algorithm for minimization of a convex quadratic function, subject to bounds on the variables, is given by Coleman and Hulbert [3].) The approach of Coleman and Li for l_1 problems [5] bears some resemblance to interior point methods, a sequence of weighted least-squares problems is solved, but it also has some distinct differences. For example, the iterates are not feasible, the l_1-function is decreased at each iteration, piecewise linear minimization is performed, and the ultimate convergence rate is quadratic.

The purpose of this paper is to propose an algorithm for l_∞-minimization that is similar in spirit to the l_1 algorithm in [5].

The l_1 algorithm proposed in [5] is a descent direction algorithm. Defining a good descent direction is nontrivial due to the hyperplanes of nondifferentiability, $a_i^T x = b_i$. The manner in which the Coleman–Li l_1 algorithm deals with nondifferentiability can be summarized as follows:

1. The l_1 algorithm generates differentiable points. Therefore, the gradient direction is defined at each iteration.
2. When far from the solution the negative gradient direction is scaled by the square root of the distances to the nondifferentiable hyperplane $a_i^T x - b_i = 0$. This is done by first globally transforming the variables so that the new variables correspond to the distances to the nondifferentiable hyperplane. This scaling helps avoid small steps.
3. Given a descent direction, lines of nondifferentiability are crossed provided the l_1 function continues to decrease.
4. Asymptotically, unit Newton steps are taken (Newton steps are defined with respect to the complementary slackness condition) thus ensuring quadratic convergence under appropriate assumptions.

The beauty of this approach is that the first-order direction and the Newton step can be combined in a *smooth and automatic* way [5].

We first introduce a few notations used in this paper. The sign function sgn(v) where v is a vector, is used in the following sense: if $w = \text{sgn}(v)$,

$$w_i = \begin{cases} 1 & \text{if } v_i \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

The residual vector is defined by $r \overset{\text{def}}{=} b - A^T x$ and $\sigma \overset{\text{def}}{=} \text{sgn}(r)$. At any point r, let $\mathcal{A}$ denote the indices of residuals with the maximum magnitude (or activities), i.e.,

$$\mathcal{A} = \left\{ l \mid |r_l| = \max_{1 \leq i \leq m} |r_i| \right\}.$$

The operator $|\cdot|$ around a set, e.g., $|\mathcal{A}|$, denotes the cardinality of this set. Otherwise it denotes the component-wise absolute values of a number, vector or a matrix. The operator $\max(x, y)$ with two vectors as arguments defines a vector whose components are the maximum of the corresponding argument vectors. The symbol $\max(x)$ of a vector x denotes the maximum component of x. The left arrow $x \leftarrow y$ denotes setting x to y. The vector e is the vector whose elements are all 1's and the vector e_j is the vector whose elements are all zero except the jth element which is equal to unity. The symbol $\overset{\text{l.s.}}{=}$ means that a linear system is solved in a least squares sense, e.g., $A^T x \overset{\text{l.s.}}{=} b$ is equivalent to solving

$$\min_{x \in \Re^n} \|A^T x - b\|_2^2.$$

In this paper we develop a new approach (similar to the Coleman and Li l_1 algorithm) for the l_∞ problem; however, this is not a trivial extension because there is no global transformation similar to that referred to above (in which the new variables themselves correspond to the distances to the planes of nondifferentiability). This is because the index j of the maximum residual, i.e., $|r_j| = \max(|r|)$, can change from iteration to iteration and the planes of nondifferentiability are defined, locally, by

$$|a_i^T x - b_i| = |a_j^T x - b_j|, \quad i \neq j, \quad i, j = 1, \cdots, m.$$

Without a global transformation the adaptation of the l_1 approach is not obvious.

In a nutshell, this problem can be overcome by using *local* transformations: at step k we define a matrix T^k that transforms the current "residuals" to variables. In this paper we show that this can be done efficiently; moreover, under appropriate assumptions, the resulting method is globally and ultimately quadratically convergent.

The linear l_∞ problem can be written as

$$\min_{x \in \Re^n} \max_{1 \leq i \leq m} |a_i^T x - b_i|. \tag{1.1}$$

The optimality conditions for (1.1) are well known: x is optimal if and only if there exists a vector μ such that

$$\sum_{i \in \mathcal{A}} a_i \sigma_i \mu_i = 0 \tag{1.2}$$

$$\sum_{i \in \mathcal{A}} \mu_i = 1, \qquad \mu_i \geq 0. \tag{1.3}$$

Note that the constraint $\sum_{i \in \mathcal{A}} \mu_i = 1$ is artificially imposed to obtain a unique definition for the multipliers μ_i: this is standard.

Let $j \in \mathcal{A}$. The optimality conditions can be (equivalently) stated:

$$\sigma_j a_j = \sum_{i \in \mathcal{A} - \{j\}} \mu_i (\sigma_j a_j - \sigma_i a_i), \tag{1.4}$$

$$\mu_i \geq 0, \qquad 1 - \sum_{i \in \mathcal{A} - \{j\}} \mu_i \geq 0. \tag{1.5}$$

DEFINITION 1. We say an l_∞ problem is *primal nondegenerate* if, at any point x, $\{\sigma_j a_j - \sigma_i a_i \mid i \in \mathcal{A} - \{j\}\}$ is of full rank for some $j \in \mathcal{A}$.

DEFINITION 2. We call λ a *dual basic point* if $A\lambda = 0$ and $|\{l : \lambda_l = 0\}| \geq m - n - 1$. We say an l_∞ problem is *dual nondegenerate* if, at any dual basic point λ, $|\{l : \lambda_l = 0\}| = m - n - 1$.

Note. If the matrix A satisfies the Haar condition, the problem is both primal and dual nondegenerate.

2. An affine scaling algorithm.

In this section we propose a new method for the l_∞-problem. This method does not possess second-order convergence; however, it is new, it is globally convergent, and it can be combined with a Newton procedure, discussed in the next section, to ultimately yield a global and second-order method (§3).

Let Z denote an $(m-n) \times m$ matrix, with rank $m - n$, such that $AZ^T = 0$. Then the l_∞ problem (1.1) is equivalent to the following constrained l_∞ problem with m

variables r:

$$\begin{aligned} &\min_{r\in\Re^m} \psi(r) \overset{\text{def}}{=} \|r\|_\infty \\ \text{(2.1)} \qquad \text{subject to} \quad & Zr = Zb. \end{aligned}$$

The objective function $\psi(r)$ is not differentiable at the points where more than one residual has maximum magnitude, i.e., $|r_l| = |r_j| = \max_{1\le i\le m} |r_i|$, $l \ne j$. However, if there is only a single maximum residual then the projected negative gradient of $\psi(r)$ onto the null space of Z is well defined and corresponds to a descent direction for $\psi(r)$. Unfortunately, it may be a poor descent direction since it can lead to a very small step if a line of nondifferentiability is immediately encountered. This can happen when there are near-activities, i.e., several residuals are nearly maximum in magnitude. Therefore, it is preferable to introduce a scaling to (partially) avoid near-active residuals. To do this we define a local transformation.

At a current point r, assume $|r_j| = \max(|r|)$ and there is no other maximum valued residual. Therefore, $\psi(r)$ is differentiable in a neighborhood of the current point r, and the nearby nondifferentiable region is (locally) defined by the hyperplanes $|r_j| - |r_i| = 0$. If we define a vector s as

$$s_i = \sigma_j r_j - \sigma_i r_i, \quad 1 \le i \le m, \quad i \ne j \tag{2.2}$$

$$s_j = \sigma_j r_j, \tag{2.3}$$

then component s_i represents the distance to the hyperplane $|r_i| - |r_j| = 0$, $i \ne j$. Alternatively, s can be written $s = |r_j|e - |r| + |r_j|e_j = T^{-1}r$, where T is a simple elementary matrix:

$$T = [-\sigma_1 e_1, \cdots, -\sigma_{j-1}e_{j-1}, \sigma, -\sigma_{j+1}e_{j+1}, \cdots, -\sigma_m e_m]. \tag{2.4}$$

Note that

$$T^{-1} = [-\sigma_1 e_1, \cdots, -\sigma_{j-1}e_{j-1}, \sigma_j e, -\sigma_{j+1}e_{j+1}, \cdots, -\sigma_m e_m].$$

Now problem (2.1) becomes

$$\begin{aligned} &\min_{s\in\Re^m} \|Ts\|_\infty \\ \text{(2.5)} \qquad \text{subject to} \quad & ZTs = Zb. \end{aligned}$$

Locally, the nondifferentiable points for $\|Ts\|_\infty$ are simply $s_i = 0$ for some i, $i \ne j$.

2.1. The search direction. Assume that r is a differentiable point and let $g = \nabla\psi(r) = \sigma_j e_j$, $D = \text{diag}(s^{1/2}) = \text{diag}((|r_j|e - |r| + |r_j|e_j)^{1/2})$, and $T = T(r)$ as defined by (2.4). We solve the following subproblem to determine a descent direction:

$$\begin{aligned} &\min_{\hat{d}_s\in\Re^m} g^T T\hat{d}_s \\ \text{(2.6)} \qquad \text{subject to} \quad & ZT\hat{d}_s = 0 \\ & \|D^{-1}\hat{d}_s\|_2 \le \delta. \end{aligned}$$

Let $\hat{d}_s = \alpha d_s$. We have

$$\begin{aligned} \text{(2.7)} \qquad d_s &= -T^{-1}A^T(AT^{-T}D^{-2}T^{-1}A^T)^{-1}Ag \\ &= -D^2T^T(g - Z^Tw^+) \end{aligned}$$

or

$$(2.8)\qquad \begin{aligned} d &= Td_s = -A^T(AT^{-T}D^{-2}T^{-1}A^T)^{-1}Ag \\ &= -TD^2T^T(g - Z^Tw^+), \end{aligned}$$

where w^+ is defined by

$$DT^TZ^Tw^+ \overset{\text{l.s.}}{=} DT^Tg.$$

So, for example, we can compute the search direction d by

$$(2.9)\qquad \begin{cases} D^{-1}T^{-1}A^Td_x \overset{\text{l.s.}}{=} DT^Tg, \\ d = -A^Td_x, \\ \lambda^+ = g + T^{-T}(D)^{-2}T^{-1}d. \end{cases}$$

Here $\lambda^+ (= Z^Tw^+)$ denotes the updated dual variables.

2.2. The algorithm. We compute d as suggested by (2.9) and then define α using a piecewise linear minimization technique along the ray d (allowing for the ability to cross lines of nondifferentiability).

Define the breakpoints α_i to be the intersections of r_i with r_j and

$$(2.10)\qquad \mathcal{J} = \left\{ \alpha_i > 0 : \ \alpha_i = -\frac{|r_j| - |r_i|}{\sigma_j d_j - \sigma_i d_i} \text{ or } \alpha_i = -\frac{|r_j| + |r_i|}{\sigma_j d_j + \sigma_i d_i} \right\}.$$

The piecewise minimization for $\psi(r + \alpha d)$ with respect to α is done by considering each positive breakpoint (intersection of a residual $|r_i + \alpha d_i|$ with the maximum residual $|r_j + \alpha d_j|$) in turn, adjusting the gradient to reflect a step just beyond the breakpoint, and then determining if d continues to be a descent direction for $\psi(r)$.

For example, let

$$(2.11)\qquad \alpha_\diamond = \min\{\alpha_i : \alpha_i \in \mathcal{J}\},$$

i.e., $\alpha_\diamond$ is the smallest positive breakpoint. Then, a step just beyond this point yields the following gradient: $g^+ = \sigma_\diamond^+ e_\diamond$. If $(g^+)^T d = \sigma_\diamond^+ d_\diamond < 0$, the intersections with $|r_\diamond|$ are considered, etc.

Asymptotically, the breakpoint will be computed by the first formula in (2.10) because the second formula in (2.10) corresponds to the case when r_i changes sign and then intersects with $|r_j|$. From (2.7), it is clear that $\alpha_i = (\sigma_i \lambda_i^+)^{-1}$. Hence the stepsize is actually determined by the dual multipliers.

The convergence of the algorithm requires the ability to cross at least one intersecting hyperplane if the current maximum residual r_j is not active at the solution. This means that the descent direction has to remain descent after crossing at least one of the intersecting hyperplanes. This is easier to manage if we can ensure that the first intersecting hyperplane is distinctly closer than the rest. However, if all multipliers have the same value at some nonoptimal vertex, then the stepsizes to the breakpoints, $(\sigma_i \lambda_i^+)^{-1}$, may become indistinguishable. The line search produces an indicator, *mod* = **true**, when the smallest positive breakpoint is jammed with the next positive breakpoint $\alpha_\ell = \min(\alpha_i : \alpha_i > \alpha_\diamond)$, i.e., if α_i and α_j are every close, then $|(\sigma_i \lambda_i^+ / \sigma_j \lambda_j^+) - 1|$ will be small.

Finally, we restrict our step to be just shy of the true minimizing point along our search direction in order to avoid a point of nondifferentiability. The parameter τ

```
Given $\tau$, mod ← false; $\alpha_\sharp^k \leftarrow 0; \sharp \leftarrow j$
Step 1. Compute the set of positive breakpoints $\mathcal{J}$ by (2.10);
Step 2. Determine the next smallest breakpoint $\alpha_l^k = \min\{\alpha_i^k : \alpha_i^k > \alpha_\sharp^k\}$. If
        $(g^+)^T d^k = \sigma_l^+ d_l^k < 0$, $\alpha_\sharp^k \leftarrow \alpha_l^k$, $\sharp \leftarrow l$, go to Step 2. Otherwise, continue;
Step 3. Let $\alpha_\ell^k = \min\{\alpha_i^k : \alpha_i^k > \alpha_l^k\}$. If $\alpha_\sharp^k = 0$ (i.e., $\alpha_\diamond^k = \alpha_l^k$) and $|\sigma_l^k \lambda_l^{k+1} -$
        $\sigma_\ell^k \lambda_\ell^{k+1}| < (1-\tau)\sigma_l^k \lambda_l^{k+1}$, set mod ← true;
Step 4. Compute the stepsize:

(2.12)          $\alpha^k = \alpha_\sharp^k + \tau(\alpha_l^k - \alpha_\sharp^k)$

        return with the value of mod and the index $l$.
```

FIG. 1. *Line search procedure* 1.

is used for this purpose; τ is typically a positive number less than but very close to unity, e.g., $\tau = .975$.

Using the information produced by the line search, we introduce another diagonal scaling matrix D_θ to ensure separation of the first positive breakpoint from the rest (thus achieving global convergence):

$$D_\theta \leftarrow \begin{cases} \mathrm{diag}(e - \frac{1}{2}e_l) & \text{if } mod = \textbf{true}, \\ \mathrm{diag}(e) & \text{otherwise.} \end{cases} \tag{2.13}$$

The effect of this perturbation is to generate a direction, in the next iteration, in which the first positive breakpoint is separated from the rest.

The (infinite) algorithm follows. In practice the loop is terminated when we are close enough to the solution: see §7 for more details.

```
Let $r^0$ satisfy $|r_i| < |r_j|, i \neq j, |r_j| = \max(|r|)$ and $Zr^0 = Zb$; $k \leftarrow 0$.

Step 1. Let $|r_j^k| = \|r^k\|_\infty$, Define $T^k = [-\sigma_1^k e_1, \cdots, -\sigma_{j-1}^k e_{j-1}, \sigma^k, -\sigma_{j+1}^k e_{j+1}, \cdots,$
        $-\sigma_m^k e_m]$, $s^k = (T^k)^{-1} r^k$, $D^k = \mathrm{diag}(s^{\frac{1}{2}}) D_\theta^k$ where $D_\theta^k$ is defined by (2.13),
        and $g^k = \sigma_j^k e_j$.
Step 2. Compute $d^k$ and $\lambda^{k+1}$ by (2.9);
Step 3. Do a line search on the piecewise linear function $\psi(r)$ along the direction $d^k$
        (Line Search Procedure 1) to determine $\alpha^k$. Update

                $r^{k+1} \leftarrow r^k + \alpha^k d^k, \qquad k \leftarrow k + 1;$
```

FIG. 2. *Algorithm* 1.

3. A local Newton process.

Close inspection of the optimality conditions for the linear l_∞ problem can yield a locally and quadratically convergent Newton process. Alternatively, consistent with the formulation in (2.1), r is optimal if and only if there exist vectors μ, w such that

$$T^T g = -\sum_{i \in \mathcal{A} - \{j\}} \mu_i e_i + T^T Z^T w, \tag{3.1}$$

$$Zr = Zb, \tag{3.2}$$

$$\mu_i \geq 0, \qquad 1 - \sum_{i \in \mathcal{A}-\{j\}} \mu_i \geq 0. \tag{3.3}$$

Recall that $g = \sigma_j e_j$. This equivalence can be seen by noticing that, using T as defined by (2.4), (1.4) can be expressed

$$\sigma_j a_j = \sum_{i \in \mathcal{A}-\{j\}} \mu_i (AT^{-T}) e_i,$$

and by comparing the above with (3.1): i.e., multiply both sides of (3.1) by AT^{-T}. Let $\lambda = Z^T w$. From (3.1),we know that if $i \in \mathcal{A}$ then $\lambda_i = \sigma_i \mu_i$; otherwise, $\lambda_i = 0$.

Now here is the key: system (3.1) can be viewed as a system of nonlinear equations. In particular, letting

$$D_r \overset{\text{def}}{=} \text{diag}(|r_j|e - |r| + |r_j|e) = \text{diag}(s),$$

the nonlinear system (3.1) is equivalent to

$$D_r T^T (g - Z^T w) = 0. \tag{3.4}$$

A (local) Newton iteration can be obtained based on the nonlinear system (3.4) and the linear constraints (3.2). However, note that T depends on the choice of the active function j; this choice need not stabilize. Nevertheless, in a neighborhood of the optimal solution (r^*, w^*), there are only a finite number of possibilities for j (there are $n+1$ activities when the problem is nondegenerate) and each nonlinear equation, corresponding to a fixed index j, has the common root (r^*, w^*). As we formally establish in §6, the local quadratic convergence behavior of a Newton process is maintained under these circumstances.

Define $D_\lambda = \text{diag}(T^T(g - Z^T w))$; differentiate (3.4), together with (3.2), to yield a Newton correction:

$$\begin{bmatrix} D_\lambda T^{-1} & -D_r T^T Z^T \\ Z & 0 \end{bmatrix} \begin{bmatrix} \triangle r \\ \triangle w \end{bmatrix} = \begin{bmatrix} -D_r T^T (g - Z^T w) \\ 0 \end{bmatrix}. \tag{3.5}$$

Define $d_N = \triangle r$; it is easy to prove that the Newton step d_N can be expressed as

$$d_N = -A^T (AT^{-T} D_r^{-1} D_\lambda T^{-1} A^T)^{-1} Ag. \tag{3.6}$$

LEMMA 1. *Assume (r^*, w^*) is the solution and the l_∞ problem is both primal and dual nondegenerate. Then there exists a neighborhood of (r^*, w^*) such that when (r, w) is within this neighborhood, and $|r_i| < |r_j| = \|r\|_\infty$, for all $i \neq j$, the matrix $AT^{-T} D_r^{-1} D_\lambda T^{-1} A^T$ is positive definite.*

Proof. The proof is similar to Lemma 1 in [5]. □

Therefore, the Newton direction becomes a descent direction for the objective function $\psi(r)$ in the neighborhood of the solution. The Newton step (3.6) is applicable in a neighborhood of the solution where it will yield quadratic convergence. In contrast, the "linear step" (2.8) is applicable globally but does not allow quadratic convergence. In the next section, we propose a new way to merge the two approaches and thereby obtain global and quadratic convergence.

4. Globalization. The linear step (2.8) can be expressed as

$$d_r = -A^T(AT^{-T}\underbrace{D_r^{-1}}T^{-1}A^T)^{-1}Ag,$$

whereas, the Newton step (3.4) equals

$$d_N = -A^T(AT^{-T}\underbrace{D_r^{-1}D_\lambda}T^{-1}A^T)^{-1}Ag.$$

The similarity in form between the linear step (2.8) and the Newton step (3.6), suggests a possible hybrid method. Our idea is to define a matrix D_θ such that the matrix $AT^{-T}D_r^{-1}D_\theta T^{-1}A^T$ goes smoothly from $AT^{-T}D_r^{-1}T^{-1}A^T$ to $AT^{-T}D_r^{-1}D_\lambda T^{-1}A^T$, as iterates converge to the solution.

Similar to [5], we define

$$\hat{D}_\theta = \text{diag}((1-\theta)|T^T(g-\lambda)| + \theta e), \tag{4.1}$$

where $\lambda = Z^T w$.

If D_λ is replaced by $\hat{D}_\theta$ above, the direction thus defined is

$$d = -A^T(AT^{-T}\underbrace{D_r^{-1}\hat{D}_\theta}T^{-1}A^T)^{-1}Ag.$$

It is clear that as $\theta \to 0$, vector d converges to a Newton step; when $\theta = 1$, d is equivalent to the step defined by (2.8).

Next we define $\theta \in [0,1]$ so that $\theta \to 0$ if and only if (r,λ) converges to a solution. We let θ encapsulate the optimality conditions. One possible choice is: Define vector v: $v_i = \max\{-\sigma_i\lambda_i, 0\}$;

$$\theta = \frac{\frac{\|D_r T^T(g-\lambda)\|_2}{\psi(r^0)} + \|v\|_\infty}{\gamma + \frac{\|D_r T^T(g-\lambda)\|_2}{\psi(r^0)} + \|v\|_\infty}, \tag{4.2}$$

where $0 < \gamma < 1$. Clearly $0 \le \theta < 1$; assuming $Z(r-b) = 0$, then $\theta = 0$ if and only if r is a solution.

If $\{r^k\}$ converges and $\bar{\theta}$ is the limit point of $\{\theta^k\}$ then for each zero residual at the limit point:

$$\alpha_i^k \to \frac{\bar{\theta} + (1-\bar{\theta})\bar{\sigma}_i\bar{\lambda}_i}{\bar{\sigma}_i\bar{\lambda}_i}, \qquad i \ne j,$$

where α_i^k corresponds to the breakpoint for residual i at the iteration k. Therefore, similar to the difficulty with the linear step, the breakpoints may not be well separated in the neighborhood of an nonoptimal point when all multipliers, corresponding to the active functions, are equal. For the same reason a perturbation is introduced in Algorithm 1, the diagonal of $\hat{D}_\theta$ needs to be further modified to help bypass nonoptimal vertices—see §2 for more discussion.

Before we discuss how to modify $\hat{D}_\theta$, we first give a line search procedure appropriate for fast convergence. In order to obtain final quadratic convergence, we must have α converge to unity sufficiently fast. Hence, instead of a constant τ, $\tau^k = \max(\tau, 1-\theta^k)$ is used to avoid nondifferentiable points. Moreover, when close to a solution, there is no need to pass as many breakpoints as possible since the breakpoints corresponding to the maximum residuals all converge to unity (see §6).

```
Given $\tau^k$, set $\sharp \leftarrow j$, $\alpha_\sharp \leftarrow 0$, $mod \leftarrow$ false
Step 1. Compute the set of positive breakpoints $\mathcal{J}$ by (2.10);
Step 2. Determine the next smallest positive breakpoint
        $\alpha_l^k$, $\alpha_l^k = \min\{\alpha_i^k : \alpha_i^k > \alpha_\sharp^k\}$;
        If we continue to descend (i.e. ${g^+}^T d^k < 0$) and we are not close to a solution
        (e.g., $\theta^k > 0.01$), $\alpha_\sharp^k \leftarrow \alpha_l^k$, go to Step 2;
        If we continue to descend but we are close to a solution (e.g., $\theta^k \le 0.01$), then
        $\alpha_\sharp^k \leftarrow \alpha_l^k$, $\alpha_l^k = \min\{\alpha_i^k : \alpha_i^k > \alpha_\sharp^k\}$, go to Step 4;
Step 3. Let $\alpha_\ell^k = \min\{\alpha_i^k : \alpha_i^k > \alpha_l^k\}$. If $\alpha_\sharp^k = 0$ ( i.e., $\alpha_l^k = \alpha_\diamond^k$ ) and $|\sigma_l^k \lambda_l^{k+1} -$
        $\sigma_\ell^k \lambda_\ell^{k+1}| < (1-\tau^k)\sigma_l^k \lambda_l^{k+1}$, set $mod \leftarrow$ true;
Step 4. Compute the stepsize:
(4.3)       $\alpha^k = \alpha_\sharp^k + \tau^k(\alpha_l^k - \alpha_\sharp^k)$.
        return with the value of $mod$ and the index $l$.
```

FIG. 3. *Line search procedure* 2.

Using the indicator *mod*, we further modify:

$$D_\theta = \begin{cases} \mathrm{diag}((1-\theta)|T^T(g-\lambda)| + \theta e - \frac{1}{2}\theta e_l) & \text{if } mod \text{ is } \textbf{true}; \\ \mathrm{diag}((1-\theta)|T^T(g-\lambda)| + \theta e) & \text{otherwise.} \end{cases} \tag{4.4}$$

A descent direction is subsequently defined using D_θ:

$$d = -A^T(AT^{-T} \underbrace{D_\tau^{-1} D_\theta} T^{-1}A^T)^{-1}Ag. \tag{4.5}$$

It is easy to prove that the matrix D_θ satisfies the following.

LEMMA 2. *Suppose* $0 < \gamma < 1$. *Assume* θ *is defined by* (4.2). *Then*

$$D_\theta \ge \tfrac{1}{2}\theta I.$$

The (infinite) algorithm follows. In §7 we give our stopping criteria.

Note. There are alternative ways to compute the search direction. For example, the following extended system can be used, provided Z is available:

$$\begin{bmatrix} D_\theta^k(T^k)^{-1} & -D_\tau^k(T^k)^T Z^T \\ Z & 0 \end{bmatrix} \begin{bmatrix} d^k \\ \Delta w^k \end{bmatrix} = \begin{bmatrix} -D_\tau^k(T^k)^T(g^k - Z^T w^k) \\ 0 \end{bmatrix}, \tag{4.6}$$

and $w^{k+1} \leftarrow w^k + \Delta w^k$, $\lambda^{k+1} \leftarrow Z^T w^{k+1}$.

5. Global convergence.

In this section we establish global convergence of the linear and hybrid methods, Algorithms 1 and 2. All the proofs are similar in spirit to those for l_1 [5]. The main difference comes from the line search technique which is reflected in Lemmas 3 and 8, and Theorem 9.

We make the following global assumption.

Assumption. The n-by-m matrix A has full row rank n.

Let $S^k = T^k D^k$ and P^k be the orthogonal projector onto null(ZS^k), i.e.,

$$P^k = I - {S^k}^T Z^T (Z S^k {S^k}^T Z^T)^{-1} Z S^k.$$

```
Let $r^0$ satisfy $|r_i| < |r_j|, i \neq j, |r_j| = \max(|r|)$; $Zr^0 = Zb$; $k \leftarrow 0$; $mod \leftarrow$ **false**;
Step 1. Let $|r_j^k| = \|r^k\|_\infty$ and
        $T^k = [-\sigma_1^k e_1, \cdots, -\sigma_{j-1}^k e_{j-1}, \sigma^k, -\sigma_{j+1}^k e_{j+1}, \cdots, -\sigma_m^k e_m]$, $s^k \leftarrow (T^k)^{-1} r^k$,
        Compute $\theta^k$ from (4.2); Set $D_\theta^k$ by (4.4);
Step 2. $D^k = (\mathrm{diag}(s^k)(D_\theta^k)^{-1})^{\frac{1}{2}}$; $g^k \leftarrow \sigma_j^k e_j$;
Step 3. Compute $d^k$ and $\lambda^{k+1}$:
            $((D^k)^{-1}(T^k)^{-1}A^T)d_x^k \overset{\text{l.s.}}{=} D^k(T^k)^T g^k$;
            $d^k \leftarrow -A^T d_x^k$;
            $\lambda^{k+1} \leftarrow g^k + T^{k^{-T}}(D^k)^{-2}(T^k)^{-1}d^k$;
Step 4. Let $\tau^k = \max(\tau, 1-\theta^k)$; Do a line search on the piecewise linear function
        $\psi(r)$ (Line Search Procedure 2) to determine $\alpha^k$:
            $r^{k+1} \leftarrow r^k + \alpha^k d^k, \qquad k \leftarrow k+1.$
```

FIG. 4. *Algorithm* 2.

The diagonal matrix $D^k = D_r^k(D_\theta^k)^{-1}$ where D_θ^k is defined by (4.4) for Algorithm 2 and D_θ^k is defined by (2.13) for Algorithm 1.

Let $g^k = \nabla\psi(r^k)$. Both algorithms generate the search direction d^k:

$$\begin{aligned} d^k &= -S^k P^k S^{k^T} g^k \\ &= -(S^k S^{k^T})(g^k - Z^T w^{k+1}) \\ &= -(TD^{k^2}T^T)(g^k - Z^T w^{k+1}), \end{aligned} \tag{5.1}$$

where w^{k+1} is the least-squares solution to

$$S^{k^T} Z^T w^{k+1} \overset{\text{l.s.}}{=} S^{k^T} g^k.$$

Alternatively, we can compute the step d by $d^k = -A^T d_x^k$, where

$$[D_\theta^k T^{-1} A^T,\ D_r^k T^T Z^T] \begin{bmatrix} d_x^k \\ w^{k+1} \end{bmatrix} = D_r^k T^T g^k. \tag{5.2}$$

The first major step in the convergence proof is to show that $\|P^k\ S^k\ g^k\| \to 0$. This is established in Lemma 6 after several preliminary results.

LEMMA 3. *Assume $\{d^k\}$ is defined by Algorithm* 1 *or Algorithm* 2. *Then*

$$\lim_{k\to\infty} \alpha_\diamond^k \|P^k S^{k^T} g^k\|_2^2 = 0, \tag{5.3}$$

where $\alpha_\diamond^k$ corresponds to the first positive breakpoint in direction d^k. Furthermore,

$$\lim_{k\to\infty} (\sigma_j^k r_j^k - \sigma_\sharp^{k+1} r_\sharp^k + \alpha_\diamond^k(\sigma_j^k d_j^k - \sigma_\sharp^{k+1} d_\sharp^k)) = 0,$$

where $\alpha_\sharp$ is defined by Line Search Procedure 1 (*or* 2) *with the convention that $\sharp = j$ if $\alpha_\sharp = 0$.*

Proof. Since $\|r^k\|_\infty$ is monotonically decreasing and bounded below, $\|r^k\|_\infty$ converges; therefore,

$$\lim_{k\to\infty}(\|r^k\|_\infty - \|r^{k+1}\|_\infty) = 0. \tag{5.4}$$

Thus

$$\begin{aligned}
\|r^k\|_\infty &-\|r^{k+1}\|_\infty \\
&= g^{k^T} r^k - g^{k+1^T} r^k - \alpha^k g^{k+1^T} d^k \qquad (\text{since } r^{k+1} = r^k + \alpha^k d^k), \\
&= (g^k - g^{k+1})^T r^k + \alpha^k (g^k - g^{k+1})^T d^k + \alpha^k \|P^k S^{k^T} g^k\|_2^2 \\
&\qquad (g^{k^T} d^k = -\|P^k S^{k^T} g^k\|_2^2) \\
&= \sigma_j^k r_j^k - \sigma_\sharp^{k+1} r_\sharp^k + \alpha^k(\sigma_j^k d_j^k - \sigma_\sharp^{k+1} d_\sharp^k) + \alpha^k \|P^k S^{k^T} g^k\|_2^2 \\
&\qquad (\text{note} : g^k = \sigma_j^k e_j) \\
&= \sigma_j^k r_j^k - \sigma_\sharp^{k+1} r_\sharp^k + \alpha_\diamond^k(\sigma_j^k d_j^k - \sigma_\sharp^{k+1} d_\sharp^k) - (\alpha^k - \alpha_\diamond^k)\sigma_\sharp^{k+1^T} d_\sharp^k + \alpha_\diamond^k \|P^k S^{k^T} g^k\|_2.
\end{aligned}$$

From

$$\sigma_j^k r_j^k - \sigma_\sharp^{k+1} r_\sharp^k + \alpha_\diamond^k(\sigma_j^k d_j^k - \sigma_\sharp^{k+1} d_\sharp^k) \geq 0,$$

we claim that

$$0 \leq \tau^k \alpha_\diamond^k \|P^k S^k g^k\|_2^2 \leq \|r^k\|_\infty - \|r^{k+1}\|_\infty. \tag{5.5}$$

To establish (5.5) note that $\sigma_\sharp^{k+1} d_\sharp^k \leq 0$. If $\alpha^k < \alpha_\diamond^k$, then $\sharp = j$ and $g^k = g^{k+1}$, $\alpha^k = \tau^k \alpha_\diamond^k$ and (5.5) holds; on the other hand, if $\alpha^k > \alpha_\diamond^k$, it is clear that again (5.5) follows.

From (5.4) and (5.5),

$$\lim_{k\to\infty} \alpha_\diamond^k \|P^k S^k g^k\|_2^2 = 0.$$

Similarly,

$$\lim_{k\to\infty}(\sigma_j^k r_j^k - \sigma_\sharp^{k+1} r_\sharp^k + \alpha_\diamond^k(\sigma_j^k d_j^k - \sigma_\sharp^{k+1} d_\sharp^k)) = 0.$$

The proof is completed. □

The proofs of Lemmas 4 and 5 are essentially identical to those of Lemmas 8 and 9 in [5]. Thus they are omitted here.

LEMMA 4. *Assume D_θ is defined by Algorithm 2 (or Algorithm 1). Under primal and dual nondegeneracy assumptions, J is nonsingular at any point $(r, \lambda = Z^T w)$, where*

$$J = \begin{bmatrix} D_\theta T^{-1} & -D_r T^T Z^T \\ Z & 0 \end{bmatrix}.$$

Moreover, $C = [D_\theta T^{-1} A^T, \ -D_r T^T Z^T]$ is also nonsingular everywhere.

LEMMA 5. *Assume that an l_∞ problem is both primal and dual nondegenerate and $\{\lambda^k = Z^T w^k\}$ is obtained by Algorithm 1 or 2. Then there exists $M > 0$ such that $\|\lambda^k\| \leq M$.*

Lemma 5 can be proved using the fact that $\{D_\theta^k\}$ is bounded above for any $\{\lambda^k\}$ and C is nonsingular everywhere (see [5, Lemma 9], for details).

We can now state the first major result.

LEMMA 6. *Assume $\{d^k\}$ is defined by Algorithm* 1 *or* 2; *assume primal and dual nondegeneracy. Then*

$$\lim_{k\to\infty} \|P^k S^k g^k\|_2 = 0.$$

Proof. Using Lemma 3, we know that

$$\lim_{k\to\infty} \alpha_\diamond^k \|P^k S^k g^k\|_2^2 = 0. \tag{5.6}$$

(Recall: $\alpha_\diamond^k$ is the step to the first positive breakpoint from r^k in the direction d^k.) From Lemma 5,there exists $M_1 > 0$, such that

$$\|(g^k - Z^T w^{k+1})\| \le M_1. \tag{5.7}$$

Now assume there exists a subsequence, which we still denote by superscript k for simplicity, satisfying

$$\{\|P^k S^k g^k\|_2^2\} \to c_1 > 0. \tag{5.8}$$

From $|d_j^k| = \|P^k {S^k}^T g^k\|_2^2 = \|S^k(g^k - Z^T w^{k+1})\|_2^2$, we have the corresponding d^k satisfying $\|d^k\|_2 \ge \frac{1}{2}c_1$ for k sufficiently large. Hence from (4.6) and Lemma 4, we know that

$$D_r^k T^T(g^k - \lambda^k) \not\to 0.$$

Thus there exists $c_2 > 0$ such that $\theta^k > c_2$.

We now prove that the corresponding subsequence of the first positive breakpoints, $\{\alpha_\diamond^k\}$, is bounded away from zero, satisfying $\alpha_\diamond^k > c_4 > 0$, and this will lead to the obvious contradiction.

From

$$\alpha_\diamond^k = -\frac{|r_j^k||{r_\diamond}^k|}{\sigma_j^k d_j^k - \sigma_\diamond^k d_\diamond^k},$$

$$\alpha_\diamond^k \ge \frac{\frac{\theta^k}{2} + (1-\theta^k)|\lambda_\diamond^k|}{(|Z_\diamond^T w^k|)} \ge \frac{c_2}{M_1} \overset{\text{def}}{=} c_4 > 0.$$

Using the facts that $\{|r_j^k|\}$ and $\{D_\theta^k\}$ are bounded away from zero and $\{|\lambda^k|\}$ is bounded above, we can similarly prove that when $\alpha_\diamond^k = -(|r_j^k| + |{r_\diamond}^k|)/(\sigma_j^k d_j^k + \sigma_\diamond^k d_\diamond^k)$, $\alpha_\diamond^k$ is also bounded away from zero, which is a contradiction.

Therefore, by (5.6), $\|P^k S^k g^k\| \to 0$. □

From Lemma 6, it follows that the point of convergence ($\bar{r}\bar{\lambda} = Z^T\bar{w}$) satisfies (3.1); this along with the nondegeneracy assumption, implies that if the primal variables converge then so do the duals. We state this formally in Lemma 7 (the proof is essentially identical to that of Lemma 11 in [5] and therefore we omit it here).

LEMMA 7. *Suppose $\{r^k\}$ and $\{w^k\}$ are obtained by Algorithm* 1 *or* 2 *and assume $\{r^k\} \to \bar{r}$, a limit point. Further, assume primal and dual nondegeneracy. Then $\{w^k\}$ converges; i.e., there exists a point $\bar{w}$ such that $\{w^k\} \to \bar{w}$. Moreover,*

$$|\mathcal{A}(\bar{r})| = n+1, \quad A\bar{\lambda} = 0, \quad \sum_{i\in\mathcal{A}} \bar{\sigma}_i\bar{\lambda}_i = 1, \quad \bar{\lambda}_i = 0 \; \forall \; i \in \bar{\mathcal{A}}^c, \; \bar{\lambda}_i \ne 0, \; \forall \; i \in \bar{\mathcal{A}}, \tag{5.9}$$

where $\bar{\lambda} = Z^T \bar{w}$.

The next result is crucial: it is established that it is not possible to have convergence to a nonoptimal point satisfying $\bar{\sigma}_j \bar{\lambda}_j < 0$ if the maximum residual j has stabilized. From this it is easy to establish (Theorem 9) that a point of convergence must be optimal. Lemma 8 applies to both Algorithm 1 and 2; however, for simplicity we give the proof only for Algorithm 2. The proof is trivially adapted to Algorithm 1: replace the definition of D_θ by (2.13).

LEMMA 8. *Assume the conditions of Lemma* 7 *are satisfied. Furthermore, assume that for all k sufficiently large the maximum residual is fixed, say function j, and $\bar{\sigma}_i \bar{\lambda}_i > 0$, $i \in \bar{\mathcal{A}} - \{j\}$. Then $\bar{\sigma}_j \bar{\lambda}_j > 0$.*

Proof. Lemma 7 immediately implies that (5.9) holds with $\bar{\lambda}_i \neq 0$ for all $i \in \bar{\mathcal{A}}$. We prove the result by contradiction.

Assume then that (5.9) holds, $\bar{\sigma}_i \bar{\lambda}_i > 0$, $i \in \bar{\mathcal{A}} - \{j\}$, but $\bar{\sigma}_j \bar{\lambda}_j < 0$. Hence $\bar{\theta} > 0$. By assumption, there exists $k_1 > 0$ such that when $k > k_1$, the index of the maximum residual is fixed. Thus, for $k > k_1$, the residual $r_\diamond$ is not crossed and

$$\alpha_l^k = \alpha_\diamond^k. \tag{5.10}$$

The index of $\alpha_\diamond^k$ is fixed for k sufficiently large and $\bar{\alpha}_\diamond < \bar{\alpha}_i$, $\bar{\alpha}_i \in \bar{\mathcal{J}}$.

By definition, a positive breakpoint α_i^k is equal to

$$\alpha_i^k = \frac{\delta_\theta^k}{|\lambda_i^{k+1}|}, \qquad i \in \bar{\mathcal{A}}, \quad i \neq j \text{ for } k \text{ sufficiently large,}$$

where $\operatorname{diag}(\delta_\theta^k) \overset{\text{def}}{=} D_\theta^k$; So ${\delta_\theta}_i^k = \theta^k + (1-\theta^k)|\lambda_i^k|$ if $i \neq l$ and

$${\delta_\theta}_l^k = \begin{cases} \theta^k + (1-\theta^k)|\lambda_l^k| & \text{if } mod = \textbf{false}, \\ \frac{\theta^k}{2} + (1-\theta^k)|\lambda_l^k| & \text{if } mod = \textbf{true}. \end{cases}$$

Hence it is clear that $\bar{\alpha}_i = \infty, i \in \bar{\mathcal{A}}^c$.

Assume for all k sufficiently large, D_θ^k is not modified, i.e., $mod =$ **false**. From the line search procedures,$|\ |\lambda_l^{k+1}| - |\lambda_\ell^{k+1}|\ | > (1-\tau)|\lambda_l^{k+1}|$ where $\ell \in \bar{\mathcal{A}}$ denotes the next positive breakpoint to the optimal. This means that $\bar{\alpha}_l < \bar{\alpha}_\ell$. Using (5.10), $\bar{\alpha}_\diamond < \bar{\alpha}_\ell$.

If, on the other hand, D_θ^k is modified an infinite number of times, we claim again that $\bar{\alpha}_\diamond < \bar{\alpha}_\ell$. Suppose $\jmath$ is such that

$$|\bar{\lambda}_\jmath| = \max_{i \in \bar{\mathcal{A}} - \{j\}} (|\bar{\lambda}_i|).$$

Let $\bar{\tau}$, $0 < \bar{\tau} < 1$, denote the limit point of $\max\{\tau, 1-\theta^k\}$, and define

$$\mathcal{E} = \left\{ i \ : \ \left| \frac{|\bar{\lambda}_\jmath|}{|\bar{\lambda}_i|} - 1 \right| \leq 1 - \bar{\tau} \right\}.$$

Then it is easy to verify that for any $i \in \mathcal{E}$

$$\frac{\bar{\theta}}{2|\bar{\lambda}_i|} + 1 - \bar{\theta} < \frac{\bar{\theta}}{|\bar{\lambda}_\jmath|} + 1 - \bar{\theta} = \bar{\alpha}_\jmath \leq \bar{\alpha}_\imath, \qquad \bar{\alpha}_\imath \in \mathcal{J}.$$

Hence, it is clear that, for k sufficiently large, if $\diamond \in \mathcal{E}$, the index of the first positive breakpoint $\alpha_\diamond^k$ remains fixed. Since D_θ^k is modified infinite number of times, then for

k sufficiently large, $\diamond \in \mathcal{E}$ and D^k_θ is modified for subsequent iterations. Therefore, $\bar{\alpha}_\diamond < \bar{\alpha}_\ell$ and the first positive breakpoint separates from the rest.

We now establish the required result: $\sigma^k_\diamond d^k_\diamond < 0$ for k sufficiently large and therefore contradicts the assumption that the maximum residual is fixed.

From (2.7), it is easy to see that $d_{si} = -s_i/\alpha_i$. Define

$$h^k_i = \frac{\alpha^k_\diamond}{\alpha^k_i} \quad \forall i \in \bar{\mathcal{A}}.$$

Hence

$$s^{k+1}_i = s^k_i - \tau^k \alpha^k_\diamond \frac{s^k_i}{\alpha^k_i} = s^k_i(1 - \tau^k h^k_i).$$

Thus, from $\bar{\alpha}_\diamond < \bar{\alpha}_i$, we have

$$\frac{1 - \tau^k}{1 - \tau^k h^k_i} < 1 - \rho, \qquad 0 < \rho < 1, \quad \text{for large enough } k \text{ and } i \neq \diamond.$$

Then

$$\frac{s^k_\diamond}{s^k_i} = \frac{s^{k-1}_\diamond}{s^{k-1}_i} \frac{1 - \tau^{k-1}}{1 - \tau^{k-1} h^{k-1}_i} < \frac{s^{k-1}_\diamond}{s^{k-1}_i}(1 - \rho) < \cdots < \frac{s^{k_2}_\diamond}{s^{k_2}_i}(1 - \rho)^{k-k_2}, \qquad i \neq \diamond.$$

Since $(1 - \rho)^{k-k_2} \to 0$, we have

$$\lim_{k\to\infty} \frac{s^k_\diamond}{s^k_i} = 0, \qquad i \neq \diamond.$$

Thus, from $d_{si} = -s_i/\alpha_i$, we have

$$\lim_{k\to\infty} \frac{d^k_{s_\diamond}}{d^k_{s_i}} = 0.$$

From

$$\bar{\sigma}_j d^k_j = \sum_{i\in\bar{\mathcal{A}}-\{j\}} \bar{\sigma}_i \bar{\lambda}_i d^k_{s_i},$$

we have

$$\bar{\sigma}_\diamond d^k_\diamond = \sum_{i\in\bar{\mathcal{A}}-\{j\}} \bar{\sigma}_i \bar{\lambda}_i d^k_{s_i} - d^k_{s_\diamond} < 0 \quad \text{for sufficiently large } k.$$

For k sufficiently large, $d^k_{s_i} < 0$ for all $i \in \bar{\mathcal{A}}$. Hence, $\sigma^k_\diamond d^k_\diamond < 0$ for k sufficiently large. Therefore, residual r_j does not remain the maximum residual, a contradiction. □

Theorem 9 below is the next major result: it says that if the primal variables converge, they converge to the optimal point.

THEOREM 9. *Assume $\{r^k\}$ is obtained from Algorithm* 1 *or* 2. *Assume primal and dual nondegeneracy. If the sequence $\{r^k\}$ converges to a point r^*, then r^* is optimal.*

Proof. Under the nondegeneracy assumption, we know that $r^*_i \neq 0, i \in \mathcal{A} = \mathcal{A}(r^*)$. From Lemma 7, $|\mathcal{A}(r^*)| = n + 1$, $\{w^k\} \to w^*$ and, if λ^* is defined by $\lambda^* =$

$Z^T w^*$, then $A\lambda^* = 0$ with $\lambda_i^* = 0$, $i \in \mathcal{A}^c$. Therefore, to establish optimality we need only show that $\sigma_i^* \lambda_i^* > 0$ for all $i \in \mathcal{A}$.

Assume the contrary, i.e., for some $i \in \mathcal{A}$, $\sigma_i^* \lambda_i^* < 0$. Then $\theta^* > 0$ and there exists k_1 such that for $k > k_1$, $r_i^k \lambda_i^k < 0$, for all i such that $r_i^* \lambda_i^* < 0$, and $r_i^k r_i^* > 0$, for all $i \in \mathcal{A}$. Thus, for $k > k_1$, $\sigma_i^{k+1} = \sigma_i^k, i \in \mathcal{A}$.

Therefore, from $d_s = T^{-1} d = -D^2 T^T (g - Z^T w^+)$, we have

$$|r_j^{k+1}| - |r_i^{k+1}| = |r_j^k| - |r_i^k| - \alpha^k {\delta_i^k}^2 (\sigma_i^k \lambda_i^{k+1}) \quad \text{for } i \in \mathcal{A},$$

where $D^k = \operatorname{diag}(\delta^k)$.

First, if there exists $k_2 > k_1$ such that $i \neq j$, i.e., $|r_i^{k_2}| < \|r^{k_2}\|_\infty$; since $\sigma_i^k \lambda_i^{k+1} < 0$ for $k > k_1$, it follows that $\|r^{k+1}\|_\infty - |r_i^{k+1}| > \|r^k\|_\infty - |r_i^k| > 0$, for all $k > k_2$. Hence, we see immediately that $\{|r_i^k|\} \not\to \|r^*\|_\infty$, which contradicts that $i \in \mathcal{A}$. Therefore, the only case $\sigma_i^* \lambda_i^* < 0$ is possible is when $i = j$ is the index of the maximum residual for all $k > k_1$ and $\sigma_i^* \lambda_i^* > 0, i \in \mathcal{A} - \{j\}$. But, by Lemma 8, this is impossible. □

Proof that our sequence converges, as stated in the following theorem, is similar to the proof of Lemma 15 and Theorem 16 in [5].

THEOREM 10. *If an l_∞ problem is both primal nondegenerate and dual nondegenerate, then the sequence $\{(r^k, \lambda^k)\}$, generated by either Algorithm* 1 *or* 2, *is convergent.*

It is now clear that under nondegeneracy assumptions, Algorithms 1 and 2 generate points that converge to the optimum point. This follows from Lemma 7 and Theorems 9 and 10.

6. Quadratic convergence. In this section we establish that Algorithm 2 produces a sequence $\{(r^k, w^k)\}$ that converges to (r^*, w^*) at a *quadratic* rate. The main difficulty is that our Newton steps come from different nonlinear systems depending on the index of the maximum residual. Similar to our approach in [5], the problem is circumvented by considering a finite set $\mathcal{F}$ of systems of nonlinear equations, where each system in $\mathcal{F}$ has the following form:

$$\left\{ \begin{array}{l} \hat{F}_j(y) \overset{\text{def}}{=} D_r T^T (g(j) - Z^T w) = 0, \\ Zr = 0, \end{array} \right. \tag{6.1}$$

where $y = [r^T, w^T]^T$, $j \in \mathcal{A}^*$ and $g(j) = \sigma_j^* e_j$. Note that T corresponds to the transformation defined by j, the maximum residual: i.e., T depends on j as well. It is trivial to see that (r^*, w^*) is a solution to each system; each system is continuously differentiable in a neighbourhood of $y^* = (r^*, w^*)$.

The Newton step at y^k, for each of the above systems F_j, is defined by

$$J_{j^k}^k d_N^k = -F_{j^k}(y^k),$$

where

$$J_{j^k}^k = \nabla F_{j^k}(y^k) = \begin{bmatrix} D_\lambda^k T^{-1} & -D_r^k T^T Z^T \\ Z & 0 \end{bmatrix}, \qquad \lambda^k = Z^T w^k,$$

and

$$F_{j^k}(y^k) = \begin{bmatrix} \hat{F}_j(y^k) \\ 0 \end{bmatrix}. \tag{6.2}$$

Note that the hybrid step d^k satisfies

$$B^k_{j^k} d^k = -F_{j^k}(y^k), \tag{6.3}$$

where

$$B^k_{j^k} = \begin{bmatrix} D^k_\theta T^{-1} & -D^k_r T^T Z^T \\ Z & 0 \end{bmatrix}, \tag{6.4}$$

and j^k corresponds to the index of the maximum residual at the iteration k. It is clear that $F_{j^k} \in \mathcal{F}$ (i.e., d^k is a Newton-like step for some $F_{j^k} \in \mathcal{F}$). Of course $F_{j^k} \neq F_{j^{k+1}}$, in general, and therefore quadratic convergence is not automatic; however, a slight modification of Theorem 3.4 in [6] yields a viable approach.

THEOREM 11. *Let $\mathcal{F} = \{F_j : \Re^m \rightarrow \Re^m\}$ be a finite set of functions satisfying:*

1. *each $F_j \in \mathcal{F}$ is continuously differentiable in an open convex set D;*
2. *there is a y^* in D such that $F_j(y^*) = 0$ and $\nabla F_j(y^*)$ is nonsingular, $F_j \in \mathcal{F}$;*
3. *$\|\nabla F_j(y) - \nabla F_j(y^*)\| \leq \eta \|y - y^*\|$ for all $y \in D$, $F_j \in \mathcal{F}$ and some $\eta \geq 0$;*

Let $\{B^k\}$ in $L(\Re^m)$ be a sequence of nonsingular matrices. Suppose that for some y^0 in D the sequence

$$y^{k+1} = y^k - (B^k)^{-1} F_{j^k}(y^k), \qquad k = 0, 1, \cdots,$$

remains in D, $y^k \neq y^*$ for $k > 0$, and converges to y^*. Moreover, assume

$$\|B^k - \nabla F_{j^k}(y^*)\| = O(\|y^k - y^*\|). \tag{6.5}$$

Then $\{y^k\}$ converges quadratically to y^*.

We now show that Algorithm 2 can be described in a manner consistent with Theorem 3 and therefore quadratic convergence is achieved. Specifically, (6.5) must be established: the next four results establish several preliminary bounds.

The next two lemma establishes that the dual multipliers and θ^k are bounded by $\|y^k - y^*\|$. The proofs are omitted because they are copies of the proofs for Lemmas 18 and 19 in [5].

LEMMA 12. *Assume that $\{(r^k, w^k)\}$ is any subsequence, convergent to (r^*, w^*), obtained by Algorithm 2. Then,*

$$\|w^{k+1} - w^k\| = \|d^k_w\| = O(\|y^k - y^*\|); \tag{6.6}$$

Consequently, $\|\lambda^{k+1} - \lambda^k\| = O(\|y^k - y^*\|)$.

LEMMA 13. *Suppose $\{\theta^k\}$ is defined as in (4.2). Then,*

$$\theta^k \leq L_1 \|y^k - y^*\|. \tag{6.7}$$

LEMMA 14. *Assume that an l_∞ problem is primal and dual nondegenerate and that the sequence $\{(r_k, w_k)\}$ is generated by Algorithm 2. Then,*

$$\alpha^k - 1 = O(\|y^k - y^*\|). \tag{6.8}$$

Proof. Since $\theta^k \rightarrow 0$, from Line Search Procedure 2, we know the stepsize $\alpha^k = \alpha^k_\sharp + (1 - \theta^k)(\alpha^k_l - \alpha^k_\sharp), \alpha^k_l > \alpha^k_\sharp$, where $l, \sharp \in \mathcal{A}^*$, or $\alpha^k_\sharp = 0$. From Lemma 12 and $\lambda^*_l \neq 0$,it is clear that

$$\alpha^k_l - 1 = \frac{\theta^k + (1 - \theta^k)|\lambda^k_l|}{|\lambda^{k+1}_l|} - 1 = O(\|y^k - y^*\|); \tag{6.9}$$

similarly,

$$\alpha_{\sharp}^k - 1 = O(\|y^k - y^*\|). \tag{6.10}$$

From Lemma 13

$$\theta^k = O(\|y^k - y^*\|);$$

therefore, using (6.9) and (6.10),

$$\alpha^k - 1 = O(\|y^k - y^*\|).$$

The proof is completed. □

Denote $B^k = B_{j^k}^k \Omega^k$ where

$$\Omega^k = \mathrm{diag}\left(\begin{bmatrix} \frac{1}{\alpha^k} e_m \\ e_{m-n} \end{bmatrix}\right)$$

and $e_m (e_{m-n})$ denotes a m-vector ($(m-n)$-vector) with each entry equal to unity, B_{j^k} is defined as in (6.4). But $y^{k+1} = y^k + \Omega^{k^{-1}} d^k$, where d^k is defined by (6.3); therefore, from Lemma 14, we have

$$\|\Omega^k - I\| = O(\|y^k - y^*\|).$$

The hybrid step defined by Algorithm 2 satisfies

$$B^k(y^{k+1} - y^k) = -F_{j^k}(y^k) \quad \text{for } k \text{ sufficiently large.}$$

LEMMA 15. *Assume* $\{y^k\}$ *is obtained through Algorithm* 2; *assume primal and dual nondegeneracy. Then*

$$\|B^k - \nabla F_{j^k}(y^*)\|_2 \le L\|y^k - y^*\|_2$$

for some $F_{j^k} \in \mathcal{F}$.

Proof. From continuity and Lemma 14, it is clear that

$$\begin{aligned} \|B^k - J_{j^k}^*\| &= \|(B_{j^k}^k - J_{j^k}^k)\Omega^k + (J_{j^k}^k - J_{j^k}^*)\Omega^k + (\Omega^k - I)J_{j^k}^*\| \\ &= O(\|B_{j^k}^k - J_{j^k}^k\|) + O(\|y^k - y^*\|) + O(\|y^k - y^*\|). \end{aligned}$$

From

$$B_{j^k}^k - J_{j^k}^k = \begin{bmatrix} (D_\theta^k - D_\lambda^k)T^{-1} & 0 \\ 0 & 0, \end{bmatrix},$$

we have

$$\begin{aligned} \|B_{j^k}^k - J_{j^k}^k\|_2 &= O(\|D_\lambda^k T^{-1} - D_\theta^k T^{-1}\|_2) \\ &= O(\|((1-\theta^k)|D_\lambda^k T^{-1}| + \theta^k I - D_\lambda^k T^{-1}\|_2) \\ &= O(\||D_\lambda^k T^{-1}| - D_\lambda^k T^{-1}\|_2) + O(\theta^k) \\ &= O(\|y^k - y^*\|_2) + O(\theta^k) \\ &= O(\|y^k - y^*\|) \qquad \text{(from Lemma 13).} \end{aligned}$$

Hence,

$$\|B^k_{j_k} - J^k_{j_k}\|_2 = O(\|y^k - y^*\|_2),$$

and therefore,

$$\|B^k - J^*_{j_k}\|_2 = O(\|y^k - y^*\|_2).$$

The proof is completed. □

The assumptions of Theorem 11 are now established; quadratic convergence of Algorithm 2 follows immediately.

THEOREM 16. *Suppose the l_∞ problem is primal and dual nondegenerate. Assume the sequence $\{(r^k, w^k)\}$ is obtained from the Algorithm 2. Then $\{(r^k, w^k)\}$ converges quadratically to (r^*, w^*).*

7. Numerical testing.

In this section we provide preliminary numerical results concerning Algorithm 2, the hybrid method (*New*). Our experiments are not exhaustive; our purpose here is to determine the viability of our approach. Is quadratic convergence observed? Is high accuracy achieved? Does the method hold promise for problems of increasing dimensions?

We have implemented the method in PRO-MATLAB [7] using SUN 3/50 and 3/160 workstations. The linear least-squares subproblems are solved using orthogonal QR-factorizations with row interchanges for greater stability [10]. No account was made of sparsity in our experiments. A starting point for the hybrid method is computed as follows[1]:

$$r^0 \leftarrow b - A^T x^0, \quad \text{where } A^T x^0 \overset{\text{l.s.}}{=} b,$$

$$\lambda^0 \leftarrow \frac{\tau}{\|r^0\|_\infty} * r^0.$$

The settings of the parameters for Algorithm 2, *New*, are

$$\tau \leftarrow .975, \qquad \gamma \leftarrow .99.$$

The dependent variable θ^k is a measure of the distance from optimality and the algorithm is stopped when

$$\theta^k < 10^{-13},$$

where machine precision on our system is approximately 10^{-16}.

We have generated two classes of test problems. First we generated random problems of varying dimensions. Second, since l_∞ minimization is often used in a function approximation context [8], we have tried several such problems.

On each test problem we compare the number of iterations to that achieved by the popular Barrodale–Phillips algorithm [1]. We do this to get a general feeling for the relative standing of the two algorithms and to examine the relative sensitivities of the two algorithms to problem size and problem class in terms of number of iterations required. We do not compare running times: we do not yet have a sparse implementation of our method.

[1] For simplicity we choose λ^0 without requiring $\lambda^0 = Z^T w^0$ for some w^0. However, the computation of λ^k for $k = 1, 2, \ldots$ ensures $\lambda^k = Z^T w^k$ for $k = 1, 2, \ldots$.

The entries in Tables 1–5 below represent the total number of required major iterations.

TABLE 1
$m = 50$

Number of Steps		
n	New	BP
10	10	24
20	9	40
30	10	47
40	9	64

TABLE 2
$m = 100$

Number of Steps		
n	New	BP
10	8	25
20	11	69
30	12	68
40	13	116
50	10	113
70	12	153
90	12	124

TABLE 3
$m = 200$

Number of Steps		
n	New	BP
10	7	25
20	12	68
30	12	113
50	13	193
70	13	249
100	13	340
140	22	382
160	15	388
190	14	302

PROBLEM 1. Random l_∞ problems: We generated the elements of matrix A and right-hand side b in a uniform random manner.

New exhibits little variation with m and n: for fixed m there is a mild increase in number of required iterations as n increases (until n gets close to m). On the other hand, *BP* requires significantly more iterations as problem size increases.

TABLE 4
$n = 5, f(z) = \exp(z)$

Number of Steps		
m	BP	New
100	10	7
200	11	8
400	12	8
600	12	8
800	12	8
1000	12	8
1200	12	9
1500	12	10
1800	12	9
2000	13	9

TABLE 5
$n = 8, f(z) = \exp(z)$

Number of Steps		
m	BP	New
100	15	7
200	18	9
400	18	10
600	21	10
800	19	9
1000	18	10
1200	22	10
1500	21	10
1800	21	11
2000	20	10

PROBLEM 2. Approximate $f(z)$, evaluated at $z = 0, \frac{1}{m}, \cdots, 1$, by a polynomial of degree $n - 1$:

$$\phi(z) = \sum_{j=1}^{n} \alpha_j \, z^{j-1},$$

where $x = (\alpha_1, \cdots, \alpha_n)$.

The relative performance of BP is much improved for approximation problems compared to random problems. As observed in [2], this is largely due to the clever starting point procedure available to the BP algorithm for approximation problems.

The quadratic convergence of the new algorithm is observed in our experiments.

8. Conclusions. In this paper we have presented a new iterative method for solving l_∞ problems. The algorithm is appealing because, similar to affine scaling approach for linear programming, the number of iterations required to solve a problem is relatively insensitive to the problem size. Moreover, since the algorithm is quadratically convergent, a solution can be obtained with high accuracy quickly (thus it is comparable to a solution obtained by a simplex type algorithm).

The algorithm is easy to implement: At each iteration, the major computation is a weighted least-squares solve. Finally, we remark that any technique available to speed up least-squares solving—e.g., exploitation of structure, sparsity, parallelism, will benefit this l_∞ algorithm directly.

REFERENCES

[1] I. Barrodale and C. Phillips, *An improved algorithm for discrete Chebychev linear approximation*, in Proc. 4th Manitoba Conf. on Numer. Math., University of Manitoba, Winnipeg, Canada, 1974, pp. 177–190.

[2] R. H. Bartels, A. R. Conn, and Y. Li, *Primal methods are better than dual methods for solving overdetermined linear systems in the l_∞ sense?*, SIAM J. Numer. Anal., 26 (1989), pp. 693–726.

[3] T. F. Coleman and L. Hulbert, *A globally and superlinearly convergent algorithm for convex quadratic programs with simple bounds*, SIAM J. Control Optim., to appear.

[4] T. F. Coleman and Y. Li, *A quadratic algorithm for the linear programming problem with lower and upper bounds*, in Large-Scale Numerical Optimization, T. F. Coleman and Y. Li, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990, pp. 49–57.

[5] ———, *A globally and quadratically convergent affine scaling method for linear l_1 problems*, Math. Programming, to appear.

[6] J. E. Dennis, Jr. and J. J. Moré, *Quasi-Newton methods, motivation and theory*, SIAM Rev., 19 (1977), pp. 46–89.

[7] C. B. Moler, J. Little, S. Bangert, and S. Kleiman, *ProMatlab User's guide*, MathWorks, Sherborn, MA, 1987.

[8] M. R. Osborne, *Finite Algorithms in Optimization and Data Analysis*, John Wiley, New York, 1985.

[9] S. A. Ruzinsky and E. T. Olsen, *l_1 and l_∞ minimization via a variant of Karmarkar's algorithm*, IEEE Trans. Acoust. Speech Signal Process., 37 (1989), pp. 245–253.

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

[11] Y. Zhang and R. A. Tapia, *A polynomial and superlinearly convergent primal-dual interior-point algorithm for linear programming*, Tech. Report 91-02, Department of Mathematical Sciences, Rice University, Houston, TX, 1991.