SIAM J. OPTIMIZATION
Vol. 3, No. 3, pp. 609–629, August 1993

010

A GLOBALLY CONVERGENT METHOD FOR l_p PROBLEMS*

YUYING LI†

Abstract. The l_p-norm discrete estimation problem $\min_{x\in\Re^n} \|b - A^T x\|_p^p$ is troublesome when p is close to unity because the objective function approaches a nonsmooth form as p converges to one. This paper presents an efficient algorithm for solving l_p-norm problems for all $1 \le p < 2$. When $p = 1$ it is essentially the method presented by T. F. Coleman and Y. Li [*Math. Programming*, 56 (1992), pp. 189–222], which is a globally and quadratically convergent algorithm under some nondegeneracy assumptions. The existing iteratively reweighted least-squares (IRLS) method can be obtained from the new approach by updating some dual multipliers in a special fashion. The new method is globally convergent, and it is superlinearly convergent when there is no zero residual at the solution. At each iteration the main computational cost of the new method is the same as that of the IRLS method: solving a reweighted least-squares problem. Numerical experiments indicate that this method is significantly faster than popular iteratively reweighted least-squares methods when p is close or equal to one.

Key words. discrete estimation, data analysis, IRLS method, linear programming, interior-point algorithm, simplex method, Newton method

AMS subject classifications. 65H10, 65K05, 65K10

1. Introduction.

In discrete estimation and data analysis it is often appropriate to solve the following problem:

$$\min_{x\in\Re^n} \|A^T x - b\|_p^p, \tag{1.1}$$

where $A = [a_1, \ldots, a_m] \in \Re^{n\times m}$, $b \in \Re^m$, and $m > n$. We denote the objective function $\|A^T x - b\|_p^p = \sum_{i=1}^m |a_i^T x - b_i|^p$ by $\psi(x)$. In this paper we focus on the case when $1 \le p < 2$. We assume that A has rank n. When $1 < p < \infty$ this assumption is equivalent to $\psi(x)$ being strictly convex. We also assume that there does not exist any x such that $A^T x - b = 0$.

The most often used measures for (1.1) are 2-norm, 1-norm, and ∞-norm. The l_∞ solution offers a worst-case guarantee. The 2-norm solution is popular because of its special relationship with the normal distributions. The increasingly important l_1 solution is useful since it is insensitive to a small number of large residuals (*resistant*). Thus one can imagine situations when minimizing the l_p-norm, where $1 < p < 2$, is appropriate [13], [14]. Moreover, the problem is theoretically interesting since it ranges from a piecewise-differentiable minimization problem (equivalent to a constrained minimization) when $p = 1$, through a once-differentiable minimization problem (but not twice differentiable) when $1 < p < 2$, to a twice-differentiable problem when $p = 2$. Clearly, it would be useful, although challenging, to develop a method that works well in all the cases.

The 2-norm problem is easy to solve: it is a simple least-squares problem. The l_1 and l_∞ problems are much more complicated and can be treated as linear programming problems and thus solved by special linear programming methods that usually take advantage of their special structures (e.g., [1], [2]). The objective function $\psi(x)$ is piecewise linear when $p = 1$ or ∞.

* Received by the editors June 20, 1991; accepted for publication (in revised form) June 6, 1992.
† Department of Computer Science, Cornell University, Ithaca, New York 14853. This research was partially supported by the U.S. Army Research Office through Mathematical Sciences Institute, Cornell University, and by the Computational Mathematics Program of the National Science Foundation grant DMS-8706133.

Until recently, the usual methods for l_1 and l_∞ have been *finite* algorithms (e.g., [1], [2]). These methods move along negative projected gradients, and the iterates tend to follow nondifferentiable hyperplanes. In contrast, Coleman and Li have recently developed *iterative* methods for l_1 and l_∞ minimization [4], [5]. These algorithms deal with the nondifferentiable hyperplanes by strategically avoiding landing on them exactly and by being able to cross when necessary. They are computationally efficient. Under a suitable nondegeneracy assumption, both algorithms proved to be globally convergent with a quadratic convergence rate.

For $1 < p < 2$ the most popular method for solving (1.1) is the iteratively reweighted least-squares (IRLS) method (e.g., [10], [13]). This method essentially takes a fixed step size along the Newton direction defined by the optimality condition $\nabla\psi(x) = 0$. It is globally linearly convergent for $1 < p < 2$ (e.g., [13]). This method can also be applied to the case $p = 1$, although no global convergence has been proved to our knowledge. With a suitable line search, the algorithm can be accelerated to be quadratically convergent when there is no zero residual at a solution. However, it is known that the method can be extremely slow, as will be further demonstrated by our numerical examples. Since the second-order derivative of the objective function does not exist when zero residuals occur, this is usually regarded as a main problem with the IRLS approach [3], [8].

The purpose of this paper is to further investigate the performance of the IRLS method and to provide a new method that works well for $1 \le p < 2$. Our experience indicates (§2) that zero residuals at a solution alone do not, in general, impede the speed of the convergence for the IRLS approach; rather, slow convergence occurs when p is close or equal to unity. This is reasonable because (1.1) is a more complicated problem (linear programming) when $p = 1$, whereas a solution for (1.1) can be obtained by solving one linear system when $p = 2$. Nonetheless, there is an additional reason for the slowness of the IRLS method: the nonlinear equation $\nabla\psi(x) = 0$ that defines the Newton step for the IRLS method does not include the conditions for a solution of (1.1) when $p = 1$. On the basis of this observation, in §3 we develop a new method by considering the system of nonlinear equations that form part of the optimality conditions for (1.1) for all $1 \le p < 2$. We also present a special line search procedure that exploits the special structure of the objective function and prevents zero residuals at each iteration. The new method performs significantly better than the IRLS methods, and it reduces to the method of Coleman and Li [5] when $p = 1$. We emphasize, however, that the main reason for the improvement is not the prevention of zero residuals in the line search. Rather, the consideration of the appropriate system of nonlinear equations brings about the improvement. In §4 we prove that the new algorithm is globally convergent. Superlinear convergence is achieved when there is no zero residual at the solution for $1 < p < 2$ and when a problem is nondegenerate for $p = 1$. Some numerical experience is reported in §5.

Finally, we introduce some notation. In this paper the superscript directly on a quantity denotes its value at the kth iteration, e.g., x^k. The conventional power operation is denoted by using brackets and superscripts together, e.g., $(x)^k$. We always use r to represent the residual vector $r = A^T x - b$, and σ denotes its sign, i.e., $\sigma = \operatorname{sgn}(r)$. Since we assume that A has full rank, the relation between x and r is a bijection. The objective function is denoted in terms of r by $\phi(r) = \|r\|_p^p, (= \psi(x))$, and the gradient $\nabla\phi(r)$, when it exists, is denoted by $g = p(|r|)^{p-1}\sigma$. In this paper 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

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

We also adopt a few MATLAB notations [11]. The symbols .* and ./ denote componentwise multiplication and division between vectors. The operator $|\cdot|$ denotes the componentwise absolute values of a number, vector, or 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 notation $\max(x)$, where x is a vector, denotes the maximum component of x, whereas the operator $\text{diag}(x)$, $x \in \Re^n$, represents the diagonal matrix with the ith diagonal element being x_i. The left arrow $x \leftarrow y$ denotes setting x to y.

2. IRLS methods. It is well known that the l_p-norm is differentiable and strictly convex for $1 < p < \infty$ under the assumption that A has full rank; thus the solution occurs at a point where the gradient $\nabla\psi(x) = Ag$ vanishes. Assume that we are at a point with $r_i \neq 0, 1 \leq i \leq m$. This is equivalent to

$$A(D)^{-2} r = 0, \tag{2.1}$$

where $D = \text{diag}((|r|)^{(2-p)/2})$. The motivation of the IRLS method comes from the fact that (2.1) forms the normal equations for the following weighted least-squares system:

$$(D)^{-1} A^T x \overset{\text{l.s.}}{=} (D)^{-1} b.$$

Thus the IRLS method can simply be described as in Fig. 1.

Given a starting point x^0

Step 1 Compute $r^k = A^T x^k - b$;

Step 2 Define $D^k = \text{diag}((|r^k|)^{(2-p)/2})$, solve x^{k+1} from

$$(D^k)^{-1} A^T x^{k+1} \overset{\text{l.s.}}{=} (D^k)^{-1} b; \qquad k \leftarrow k + 1; \tag{2.2}$$

Go to Step 1;

FIG. 1. IRLS *Algorithm.*

Remark. In the description of the algorithm it is implicitly assumed that at each iteration $r_i^k \neq 0$, $1 \leq i \leq m$. In practice, care must be taken when some $r_i^k = 0$. Let $e^T = [1, \ldots, 1] \in \Re^m$ be the m vector of all ones. Watson [14] suggested using $D^k = \text{diag}(\max\{\delta e, |r^k|\})^{(2-p)/2}$ for some small positive constant δ. In our implementation we use $D^k = \text{diag}(100\epsilon e + (|r^k|)^{(2-p)/2})$, where ϵ is the machine precision.

Let us inspect the step $x^{k+1} - x^k$ taken by IRLS more closely. If it is assumed that $r_i^k \neq 0 \ \forall 1 \leq i \leq m$, a Newton step in x-space for (2.1) can be defined through differentiation as

$$d_N^k = -\frac{1}{p-1}(A(D^k)^{-2}A^T)^{-1} A(D^k)^{-2} r^k. \tag{2.3}$$

It is clear that d_N^k is always a descent direction for the objective function $\psi(x)$. Consider the increment $\Delta x^k = x^{k+1} - x^k$ obtained from IRLS:

$$\begin{aligned}\Delta x^k &= (A(D^k)^{-2}A^T)^{-1}A(D^k)^{-2}b - (A(D^k)^{-2}A^T)^{-1}A(D^k)^{-2}A^T x^k \\ &= -(A(D^k)^{-2}A^T)^{-1}A(D^k)^{-2}r^k \qquad (p \geq 1) \\ &= (p-1)d_N^k \qquad (p > 1).\end{aligned}$$

Hence Δx^k can be considered as a damped Newton step (e.g., [14]).

In [13] it is proved by assuming $1 < p < 2$ and $r_i^k \neq 0$, $1 \leq i \leq m$, that the limit point of the sequence $\{x^k\}$ generated by the IRLS algorithm is a solution to (1.1) and that the convergence is linear with convergence constant $2 - p$. Wolfe [15] obtained the same local convergence property with a rather involved proof.

When $p = 1$ there is no global convergence result to our knowledge; however, if global convergence is assumed, then the convergence rate will be linear [13]. We claim that a slight modification of the proofs in [5] yields that IRLS, when $p = 1$, is also globally convergent under some nondegeneracy assumptions.

The above IRLS method has a linear convergence rate because of its failure to take a full Newton step. However, taking a full Newton step at each iteration may lead to divergence [10]. Nonetheless, a line search globalization of the Newton method can be made to achieve final quadratic convergence and to maintain global convergence at the same time.

In this paper a line search procedure is used for improving both the IRLS method and the new algorithm. We refer to the modified IRLS algorithm as IRLSL (IRLS with the line search). Note that d_N^k as defined in (2.3) is the solution to the following least-squares problem:

$$(D^k)^{-1}A^T d_x^k \overset{\text{l.s.}}{=} -D^k g^k, \quad \text{where } D^k = (\operatorname{diag}(|r^k|)(\operatorname{diag}(p-1)|g^k|)^{-1})^{1/2}.$$

A model algorithm for IRLSL in terms of r is described in Fig. 2.

Given an initial point $r^0 = A^T x^0 - b$ *with* $|r^0| > 0$

Step 1 Compute $g^k = p(|r^k|)^{p-1}\sigma^k$, $D_r^k = \operatorname{diag}(|r^k|)$ and $D^k = (D_r^k \operatorname{diag}((p-1)|g^k|)^{-1})^{1/2}$;

Step 2 Compute the direction d^k by solving

$$\text{(2.4)} \qquad \begin{cases} (D^k)^{-1}A^T d_x^k \overset{\text{l.s.}}{=} -D^k g^k \\ d^k = A^T d_x^k; \end{cases}$$

Step 3 Perform the line search as described below (see Fig. 3). Update

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

Go to Step 1;

FIG. 2. IRLSL *Algorithm.*

Remark. We describe IRLSL in this fashion so as to compare it with our new algorithm, which will be presented in §3. Since $r = A^T x - b$ and A has full rank, x can be recovered from r on termination if needed. Alternatively, one can choose to update x directly at each iteration.

Now we discuss how to determine a suitable step size α^k. Given any descent direction $d^k \in \Re^m$, we determine a suitable step size α^k by attempting to minimize $\phi(r^k + \alpha d^k)$ over $\alpha \geq 0$.

The objective function $\phi(r)$ is continuously differentiable when $p > 1$. Thus, by following [7, Thm. 6.3.2], given $0 < \beta_f < \beta_g < 1$, there exists $0 < \alpha_l^k < \alpha_u^k$ such that, when $\alpha^k \in [\alpha_l^k, \alpha_u^k]$, the following conditions are satisfied at $r^{k+1} = r^k + \alpha^k d^k$:

$$\phi(r^{k+1}) \leq \phi(r^k) + \beta_f \alpha^k \nabla\phi(r^k)^T d^k, \tag{2.5}$$

$$\nabla\phi(r^{k+1})^T d^k \geq \beta_g \nabla\phi(r^k)^T d^k. \tag{2.6}$$

Unfortunately, the objective function $\phi(r)$ is not twice differentiable everywhere. Hence conditions (2.5) and (2.6) do not guarantee convergence to the solution. The difficulty is that condition (2.6) may not guarantee large enough step lengths because of the nonsmoothness of the derivatives.

Since the function $\phi(r^k + \alpha d^k)$ is strictly convex (under our assumption), there can be only one minimum along d^k. However, we do not want to perform an exact line search when $p > 1$ because of concern for efficiency. Instead, we exploit the special structure of the objective function $\phi(r)$ and perform the line search in the following fashion.

Consider the following strictly convex quadratic function $U^k(r)$ around a differentiable point r^k, i.e., $r_i^k \neq 0$, $1 \leq i \leq m$, as defined by Osborne [13, p. 252]:

$$U^k(r) = \frac{1}{2}\sum_{i=1}^{m} \frac{|g_i^k|}{|r_i^k|}(r_i)^2 + \sum_{i=1}^{m}\left((|r_i^k|)^p - \frac{1}{2}\frac{|g_i^k|}{|r_i^k|}(|r_i^k|)^2\right), \tag{2.7}$$

where $g^k = \nabla\phi(r^k)$. This quadratic function has been used [13, p. 252] to prove that $\{r^k\}$ generated by IRLS decreases the objective function $\phi(r)$ monotonically. It has the following properties [13, p. 252]:

$$\phi(r) \leq U^k(r), \qquad \phi(r^k) = U^k(r^k), \tag{2.8}$$

and

$$\nabla U^k(r^k) = \nabla\phi(r^k), \quad \nabla^2 U^k(r) = \operatorname{diag}(p(|r^k|)^{p-2}).$$

Moreover, $\arg\min_x U^k(r) = x^{k+1}$, where x^{k+1} is defined by the IRLS algorithm (cf. (2.2)). In other words, $U^k(r)$ is a special quadratic interpolation of $\phi(r)$.

In this paper we use this quadratic interpolation to facilitate the line search in both IRLSL and in our new method, which will be discussed later. We calculate the minimizer of $U^k(r)$ along any descent direction d^k and use it to approximate the minimizer of $\phi(r)$ along this direction. The minimizer for $U^k(r^k + \alpha d^k)$ equals

$$\check{\alpha}^k = -\frac{g^{k^T} d^k}{d^{k^T} \operatorname{diag}(p(|r^k|)^{p-2}) d^k}. \tag{2.9}$$

For IRLSL $d^k = A^T d_x^k$, where d_x^k is defined by (2.4). Thus $\check{\alpha}^k = p - 1$ and $\check{\alpha}^k$ is the step size that IRLS takes at each iteration when $p > 1$.

The following lemma indicates that for any descent direction d^k, $\check{\alpha}^k$ always introduces a sufficient decrease in objective function value. This explains why the sequence generated by IRLS converges to a solution when $1 < p < 2$.

LEMMA 2.1. *Assume* $1 \le p \le 2$. *Given any descent direction* d^k, *the step size* $\check{\alpha}^k$ *as defined by* (2.9) *satisfies* (2.5) *with any* $0 \le \beta_f \le \frac{1}{2}$.

Proof. Let U^k be as defined by (2.7). Then

$$\begin{aligned}
&\phi(r^k) - \phi(r^k + \check{\alpha}^k d^k) \\
&\quad \ge U^k(r^k) - U^k(r^k + \check{\alpha}^k d^k) && \text{(from (2.8))} \\
&\quad = -\check{\alpha}^k g^{k^T} d^k - \frac{1}{2}(\check{\alpha}^k)^2 d^{k^T} \mathrm{diag}(p(|r^k|)^{p-2}) d^k && (U^k(r) \text{ is a quadratic}) \\
&\quad = -\check{\alpha}^k g^{k^T} d^k + \frac{1}{2}\check{\alpha}^k g^{k^T} d^k && \text{(from (2.9))} \\
&\quad = -\frac{1}{2}\check{\alpha}^k g^{k^T} d^k \\
&\quad \ge -\beta_f \check{\alpha}^k g^{k^T} d^k.
\end{aligned}$$

Hence (2.5) is satisfied. □

Quadratic interpolation techniques have been used in line search methods for general nonlinear minimization [9]. However, it is worth emphasizing that for general nonlinear functions the interpolation function is usually a one-dimensional function that is defined along a search direction instead of approximating the objective function in the entire space. For any given problem (1.1), the interpolation function $U^k(r)$ used here guarantees that the step size $\check{\alpha}^k$ is acceptable for $\phi(r)$ (i.e., sufficient decrease is achieved and the step is not too small), which usually cannot be achieved for general nonlinear functions.

Since $\phi(r)$ becomes increasingly close to being nondifferentiable as p gets close to one, $\check{\alpha}^k$ may be a bad choice (it converges to zero). When the objective function $\phi(r)$ is not differentiable, i.e., $p = 1$, the exact minimizer $\phi(r^k + \alpha d^k)$ occurs at a nondifferentiable point. Along any direction $d^k \in \Re^m$ the points at which the second-order derivatives fail to exist can easily be calculated. We refer to the step sizes corresponding to such points as breakpoints. The set $\mathcal{J}$ identifies the positive breakpoints:

$$\mathcal{J} = \left\{ \alpha_i^k : \ \alpha_i^k = -\frac{r_i^k}{d_i^k}, \ r_i^k d_i^k < 0 \right\}. \tag{2.10}$$

The basic idea behind our line search procedure is to take larger step sizes when possible. We consider the first positive breakpoint at which d^k becomes an ascent direction, a unit step size, and $\check{\alpha}^k$ in this order. The first at which the objective function is sufficiently decreased (i.e., (2.5) is satisfied) is accepted.

However, the exact nondifferentiable point needs to be avoided (if a unit step size or $\check{\alpha}$ is taken, it is unlikely that r^{k+1} is a nondifferentiable point). We achieve this by slightly stepping back from a nondifferentiable point. Assume that $r_i^k + \omega d_i^k = 0$ at the step size $\omega > 0$ under consideration. Let

$$\alpha_\sharp^k \leftarrow \max\{\alpha_i^k : 0 \le \alpha_i^k < \omega\},$$

and set

$$\alpha^k \leftarrow \alpha_\sharp^k + \tau^k(\omega - \alpha_\sharp^k), \tag{2.11}$$

where $\tau^k \in (0, 1)$.

For IRLSL we choose

$$\tau^k = \max\left(\tau, 1 - \frac{\|Ag^k\|_2}{1 + \|Ag^k\|_2}\right) \tag{2.12}$$

and $\tau \in (0,1)$, e.g., $\tau = 0.975$. When $1 < p < 2$, $\|Ag^k\|$ is a measure of optimality. When a solution for $1 < p < 2$ is approached $\|Ag^k\|$ converges to zero and thus τ^k converges to one. Hence if (2.5) is satisfied with the unit step size, the perturbed α^k converges to unity, which is required for fast local convergence. By assuming that (2.5) is satisfied with ω, it is easy to verify that (2.5) is also satisfied with α^k defined by (2.11) since $\phi(r)$ is convex under our assumption. Thus (2.5) is always satisfied for the step size computed.

The line search procedure is summarized in Fig. 3. We point out that when $p = 1$, $g(r^k + \alpha_*^k d^k)$ does not exist and the gradient just past the breakpoint α_*^k is used (for details see [5]). Moreover, if it is assumed that $\beta_f = 0$ and $p = 1$, this line search procedure always locates the exact minimizer and ensures that $r_i^k \neq 0, 1 \leq i \leq m$, at each iteration (the line search procedure always returns at Step 1).

Given $\tau^k, \beta_f \in (0,1)$, d^k, r^k, $\check{\alpha}^k$, α_i^k (*defined by* (2.10)), *and a large* $\rho_B > 0$ (*e.g.,* 10^6)

Step 1 Let α_*^k be the smallest positive breakpoint in $[\check{\alpha}^k, \rho_B]$ with $g(r^k + \alpha_*^k d^k)^T d^k \geq 0$. If such a breakpoint α_*^k exists and (2.5) is satisfied with α_*^k, let $\alpha_\sharp^k \leftarrow \max\{\alpha_i^k : 0 \leq \alpha_i^k < \alpha_*^k\}$ and set

$$\alpha^k \leftarrow \alpha_\sharp^k + \tau^k(\alpha_*^k - \alpha_\sharp^k)$$

and return; Otherwise, continue;

Step 2 If (2.5) is not satisfied with $\alpha^k = 1$, continue to the next step. Otherwise, set

$$\alpha^k \leftarrow \begin{cases} 1 & \text{if } \min(|r^k + d^k|) > 0, \\ \alpha_\sharp^k + \tau^k(1 - \alpha_\sharp^k) & \text{otherwise,} \end{cases}$$

where $\alpha_\sharp^k \leftarrow \max\{\alpha_i^k : 0 \leq \alpha_i^k < 1\}$, return;

Step 3 Set

$$\alpha^k \leftarrow \begin{cases} \check{\alpha}^k & \text{if } \min(|r^k + \check{\alpha}^k d^k|) > 0, \\ \alpha_\sharp^k + \tau^k(\check{\alpha}^k - \alpha_\sharp^k) & \text{otherwise,} \end{cases}$$

where $\alpha_\sharp^k \leftarrow \max\{\alpha_i^k : 0 \leq \alpha_i^k < \check{\alpha}^k\}$, return;

FIG. 3. *Line search procedure.*

In §4 we will prove that the IRLSL algorithm with the above line search procedure is globally convergent. It is quadratically convergent when there is no zero residual at the solution.

Merle and Späth [10] empirically studied the IRLS algorithm and concluded that the IRLS algorithm (without a line search) is satisfactory. We disagree with this claim. To investigate the performance of the algorithms more carefully, we apply both the IRLS and IRLSL algorithms to some randomly generated l_p-norm problems

(for details, see §5). The following stopping criterion is used:

$$\textbf{either} \quad \left|\frac{\psi(x^k) - \psi(x^{k+1})}{\psi(x^k)}\right| < \tau_s = \frac{1}{2} \times 10^{-11} \quad \textbf{or} \quad \text{itcount} > 50.$$

Here itcount denotes the number of iterations. For more discussion of the stopping criterion, see §5.

TABLE 1
Behavior of the two algorithms when p approaches one.

Number of Iterations ($m = 100$, $n = 50$)											
$p =$	1	1.01	1.02	1.03	1.04	1.05	1.06	1.07	1.08	1.09	1.1
IRLSL	50	50	36	50	40	38	28	33	30	29	24
IRLS	50	50	50	50	50	50	50	50	50	50	50

TABLE 2
Effect of zero residuals.

Number of Iterations ($m = 100$, $n = 50$)					
p (no $r_i^* = 0$)	IRLS	IRLSL	p (five $r_i^* = 0$)	IRLS	IRLSL
1.3	32	14	1.3	31	18
1.4	23	8	1.4	22	12
1.5	17	9	1.5	17	11
1.6	13	7	1.6	13	8
1.7	10	7	1.7	10	7
1.8	8	5	1.8	8	7
1.9	6	5	1.9	6	5

Tables 1 and 2 represent typical performance of IRLS and IRLSL.

First, we observe that IRLSL is more efficient than IRLS. Our computational experience indicates that IRLSL converges faster than does IRLS (e.g., Tables 1 and 2): even when both methods fail to find a solution, IRLSL computes an approximate solution with a lower objective function value. The additional cost per iteration for IRLSL is that of the line search, which is roughly $O(\kappa m)$, where κ is the number of positive breakpoints in $[\check{\alpha}, \rho_B]$ that have to be inspected in order to find α_*, i.e., an inner product needs to be computed at every such point. In our experiments this number κ is in general much less than n and decreases quickly as p departs from unity. Thus the cost of the line search is of a lower order than that of solving a least-squares problem ($O(mn^2)$). Hence we conclude that IRLSL is more efficient than IRLS, and subsequently we will compare our new method (§3) with IRLSL only.

As indicated in Table 1, both algorithms (with or without a line search) converge increasingly slowly when p approaches unity. It is clear that when $p = 1$ one can always find a solution with n zero residuals (e.g., [1]). Thus when p is close to unity, there usually exist either zero residuals or extremely small residuals. Because the Hessian matrix of the objective function does not exist at points with zero residuals when $1 < p < 2$, it seems to be reasonable to blame the slow convergence on the occurrence of the zero residuals at a solution.

However, we argue that this is not the reason. Our argument is supported by the results in Table 2, which indicate that the presence of zero residuals at solutions does not significantly affect the algorithm when p is further away from unity. When a random l_p-norm problem, $p > 1.5$, is generated, it usually does not have zero residuals

at the solution. For comparison, we generate random l_p-norm problems in a special way to guarantee the zero residuals at the solution: we solve an l_p-norm problem first and add more residuals so that they equal zero at the solution. As indicated by Table 1, both algorithms seem to be unaffected by the presence of zero residuals at a solution.

We will further investigate this question in the next section.

3. A new algorithm. The IRLSL method works well when p is sufficiently far from unity (e.g., $p > 1.3$), as indicated by our numerical results. However, when p is close to unity it becomes unsatisfactory. Moreover, our numerical experience indicates that a zero residual does not necessarily impede the speed of convergence. Hence alternative reasons for the slowness of the IRLS methods must be sought.

Recall that the descent directions used by both IRLS and IRLSL are derived from the nonlinear equations $\psi(x) = 0$. This is the optimality condition for (1.1) when $1 < p < 2$ but not when $p = 1$. Hence when $p = 1$ slow progress is made by moving along these descent directions because no attempt is made to satisfy the optimality conditions directly. We believe that this is the cause of the unsatisfactory performance of the IRLS methods when p is close or equal to unity.

Let the rows of the matrix Z form a basis for the null space of A, i.e., $AZ^T = 0$. Recall that $g = p(|r|)^{p-1}\sigma$. We can write (2.1) in the following equivalent form:

$$g - Z^T w = 0. \tag{3.1}$$

The number of equations is m, which is equivalent to the number of variables (x, w) (note that $x \in \Re^n$ and $w \in \Re^{m-n}$).

Let $D_r^k = \text{diag}(|r^k|)$, and denote $\lambda^k = Z^T w^k$. At any point (x^k, w^k) the Newton step for the above equations is defined by

$$\left[p(p-1)\text{diag}((|r^k|)^{p-2})A^T, -Z^T\right]\begin{bmatrix} d_x^k \\ d_w^k \end{bmatrix} = -\left[g^k - \lambda^k\right]. \tag{3.2}$$

Thus the Newton step for the x variables is

$$d_x^k = -\frac{1}{p-1}(A(D_r^k)^{-1}\text{diag}(|g^k|)A^T)^{-1}Ag^k,$$

which is equivalent to the Newton step (2.3) for $\nabla\psi(x) = 0$.

Now we consider the following nonlinear system of equations:

$$D_r(g - Z^T w) = 0. \tag{3.3}$$

When $p = 1$ this is the complementary slackness condition for a solution and λ is often called the vector of *dual multipliers*. In [5] we have used (3.3) to define local Newton steps for l_1-norm problems. When $1 < p < 2$, (3.3) is the optimality condition for (1.1) if D_r is nonsingular.

A solution to (3.1) is always a solution to (3.3). A solution (x, w) to (3.3) is a solution to (3.1) if for any $r_i = 0$, $\lambda_i = 0$. Hence we can compute a solution of (1.1) by satisfying (3.3) and the condition that $\lambda_i = 0$ if $r_i = 0$.

By considering (3.3) instead of (3.1), we capture both the optimality conditions for smooth minimization ($p > 1$) and part of the optimality conditions for nonsmooth minimization ($p = 1$). Given that the objective function $\phi(r)$ becomes nearly nonsmooth when p is close to unity, we argue that it is better to consider (3.3) than to

consider (3.1). Since we are concerned with the l_p-norm problem for all $1 \leq p < 2$, taking Newton steps defined by (3.3) is more appropriate than using (3.1). This is the main idea behind our new method. Next we describe our new method in more detail.

Assume for now that the Jacobian of $D_r(g - Z^T w)$ exists at (x^k, w^k) and is nonsingular. Let $D_\lambda^k = \text{diag}(p\sigma^k .* g^k - \sigma^k .* \lambda^k)$. Then the Newton step for (3.3) is defined by

$$\left[D_\lambda^k A^T, -D_r^k Z^T\right] \begin{bmatrix} d_x^k \\ d_w^k \end{bmatrix} = -\left[D_r^k (g^k - \lambda^k)\right]. \tag{3.4}$$

Hence we obtain

$$A(D_r^k)^{-1} D_\lambda^k A^T d_x^k = -Ag^k \tag{3.5}$$

or, equivalently,

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

In [5] we have proved that when $p = 1$, $A(D_r)^{-1} D_\lambda A^T$ is positive definite in the neighborhood of the solution, under some nondegeneracy assumptions.

Consider the case in which $1 < p < 2$. If there is no zero residual at the solution, i.e., $|r^*| > 0$, $(D_r^*)^{-1} D_\lambda^*$ is positive definite since $D_\lambda^* = (p-1)\text{diag}(|g^*|)$ and A is assumed to have full rank. Thus $A(D_r^k)^{-1} D_\lambda^k A^T$ is also positive definite when (x^k, w^k) is close to (x^*, w^*). Hence the Newton direction d_x^k becomes a descent direction for $\psi(x)$ in a neighborhood of the solution.

If there exists some $r_i^* = 0$, the Jacobian matrix of (3.3) is singular at the solution when $1 < p < 2$ because $g_i^* = \lambda_i^* = 0$. However, at those points the Jacobian matrix of the original system (2.1) does not exist either. Hence this trouble is not introduced by considering (3.3) instead of (3.1). If there exists a zero residual at a solution x^*, it is difficult to achieve quadratic convergence and we are content with fast linear convergence.

Since $A(D_r^k)^{-1} D_\lambda^k A^T$ may not be positive definite far from a solution, globalization of the Newton step (3.5) is required.

First, we recall the technique used in [5] for $p = 1$. In [5] the Newton method is globalized by defining a diagonal matrix D_θ^k such that $A(D_r^k)^{-1} D_\theta^k A^T$ changes from $A(D_r^k)^{-1} A^T$ to $A(D_r^k)^{-1} D_\lambda^k A^T$ as the solution is approached and by replacing D_λ^k by D_θ^k when a direction is computed by (3.4). Thus the hybrid step can be considered as the solution to the following linear equations:

$$\left[D_\theta^k A^T, -D_r^k Z^T\right] \begin{bmatrix} d_x^k \\ d_w^k \end{bmatrix} = -D_r^k (g^k - \lambda^k). \tag{3.7}$$

Hence

$$d_x^k = -(A(D_r^k)^{-1} D_\theta^k A^T)^{-1} Ag^k. \tag{3.8}$$

If a controlling variable $0 < \theta < 1$ that measures the closeness to the solution is used, the diagonal matrix D_θ is defined in the following way:

$$D_\theta^k = |\theta^k \text{diag}(\sigma^k g^k) + (1 - \theta^k) D_\lambda^k| = \text{diag}(|g^k - (1 - \theta^k)\lambda^k|). \tag{3.9}$$

Here θ^k measures the satisfaction of the complementary slackness condition and the dual feasibility of an l_1-norm problem

$$\theta^k = \frac{\eta^k}{\gamma + \eta^k},$$
$$\eta^k = \max\left\{\max\left\{\frac{|D_r^k(g^k - \lambda^k)|}{\phi(r^0)}\right\}, \max\{\max\{|\lambda^k| - |g^k|, 0\}\}\right\}, \tag{3.10}$$

and $0 < \gamma < 1$ (in our implementation $\gamma = 0.99$). In other words, η^k is the maximum of the violation of the complementary slackness condition ($D_r(g - \lambda) = 0$) and of dual feasibility ($|\lambda| \leq |g|$). Note that $|g| = |p(|r|)^{p-1}| = e$ when $p = 1$. In this case, $\theta = 0$ (or $\eta = 0$) is a necessary and sufficient condition of optimality. (For a more detailed discussion see [5].)

Now we consider the case in which $1 < p < 2$. Since we know that the direction defined by the IRLS methods leads to global convergence, we want to define a diagonal matrix D_θ such that globally the direction defined by replacing D_λ by D_θ is the same direction as that of IRLS and that locally it converges to D_λ. Notice that if we let $D_\theta = |\text{diag}((p-1)g)|$, the direction defined by (3.7) equals the IRLS direction (2.3). Unfortunately, a simple scalar combination $|\theta\,\text{diag}(pg) + (1-\theta)D_\lambda|$ does not lead to the IRLS direction globally because some components of the combination may not approach zero when the corresponding components in $\text{diag}(p|g|)$ converge to zero. We form the diagonal matrix D_θ in a slightly more complicated way: the diagonal is the componentwise convex combination of that of $\text{diag}(pg^k)$ and D_λ^k:

$$\begin{aligned} D_\theta^k &= |\text{diag}(\theta^k)\text{diag}(p\sigma^k g^k) + \text{diag}(e - \theta^k)D_\lambda^k| \\ &= \text{diag}(|pg^k - (e - \theta^k) .*\ \lambda^k|). \end{aligned} \tag{3.11}$$

(Recall that the operators .* and ./ denote componentwise multiplications and divisions between vectors.) Here θ^k is a vector

$$\theta^k = (\eta^k e) ./ (\gamma|g^k| + \eta^k e), \tag{3.12}$$

where γ is, again, a constant with $0 < \gamma < 1$ and $e^T = [1, \ldots, 1] \in \Re^m$. The scalar η^k is as defined in (3.10). It is clear that when $p = 1$, (3.11) is the same as definition (3.9), which is used in [5]. Hence when $p = 1$, D_θ^k defined by (3.12) is equivalent to that defined by (3.10). Moreover, x is optimal if and only if there exists $\lambda = Z^T w$ such that $\eta = 0$.

The diagonal matrix D_θ^k has the following properties.

LEMMA 3.1. *Suppose* $0 < \gamma < 1$. *Assume* D_θ^k *is defined by* (3.11). *Then* D_θ^k *satisfies*

$$(p-1)\text{diag}(|g^k|) \leq |D_\theta^k| \leq (p+1)\text{diag}(|g^k|). \tag{3.13}$$

Proof. By definition (3.11)

$$D_\theta^k = \text{diag}(|pg^k - (e - \theta^k).*\lambda^k|).$$

From definition (3.12) of θ

$$\eta^k(e - \theta^k) = \gamma\theta^k .* |g^k|.$$

Hence

$$(|\lambda^k| - |g^k|).*(e-\theta^k) \le \gamma\theta^k.*|g^k|.$$

Therefore,

$$\begin{aligned}|\lambda^k| &\le |g^k| + \gamma(\theta^k.*|g^k|)./(e-\theta^k)\\ &\le ((e-\theta^k).*|g^k| + \gamma\theta^k.*|g^k|)./(e-\theta^k)\\ &\le ((e-(1-\gamma)\theta^k).*g^k)./(e-\theta^k).\end{aligned}$$

Hence

$$(p-1)\mathrm{diag}(|g^k|) \le |D_\theta^k| \le (p+1)\mathrm{diag}(|g^k|). \qquad \square$$

As will be shown in §4, with a suitable line search this globalization guarantees that when $1 < p < 2$, $\{\lambda_i^k\}$ converges to zero if $\{r_i^k\}$ converges to zero. Hence the corresponding $\{x^k\}$ converges to a solution of (1.1).

We apply the same line search procedure to the new method. However, the definition of τ^k in (2.12) is replaced by

$$\tau^k = \max\left(\tau, 1 - \frac{\eta^k}{\gamma+\eta^k}\right) \tag{3.14}$$

so as to include the measure of the optimality for $p = 1$. Note that when $\{\eta^k\}$ converges to zero, $\{\tau^k\}$ converges to unity. When $p = 1$ the line search procedure for the new method is equivalent to the one used in [5].

For IRLSL $\check{\alpha}^k$ is a constant $p-1$. For our new algorithm with d^k defined by (3.8), $\check{\alpha}^k$ changes at each iteration. However, it is bounded between $p-1$ and $p+1$ as indicated by the following lemma.

LEMMA 3.2. *Assume $d = A^T d_x^k$, where d_x^k is defined by* (3.8). *Then the step size $\check{\alpha}^k$ as defined by* (2.9) *satisfies*

$$p - 1 \le \check{\alpha}^k \le p + 1.$$

Proof. By definition (2.9)

$$\begin{aligned}\check{\alpha}^k &= -\frac{{g^k}^T d^k}{{d^k}^T \mathrm{diag}(p(|r^k|)^{p-2})d^k}\\ &= \frac{{d^k}^T (D_r^k)^{-1} D_\theta^k d^k}{{d^k}^T \mathrm{diag}(p(|r^k|)^{p-2})d^k} \qquad \text{(from (3.8))}.\end{aligned}$$

From (3.13)

$$(p-1)\frac{{d^k}^T (D_r^k)^{-1}\mathrm{diag}(|g^k|)d^k}{{d^k}^T \mathrm{diag}(p(|r^k|)^{p-2})d^k} \le \check{\alpha}^k \le (p+1)\frac{{d^k}^T (D_r^k)^{-1}\mathrm{diag}(|g^k|)d^k}{{d^k}^T \mathrm{diag}(p(|r^k|)^{p-2})d^k}.$$

This means

$$(p-1) \le \check{\alpha}^k \le (p+1). \qquad \square$$

Computationally, instead of solving an $m \times m$ linear system (3.7) to compute (d_x^k, d_w^k), one may prefer to compute d_x^k by solving an $m \times n$ least-squares problem

$$(D^k)^{-1} A^T d_x^k \overset{\text{l.s.}}{=} -D^k g^k,$$

where $D^k = (D_r^k (D_\theta^k)^{-1})^{1/2}$. Hence

$$\begin{cases} (D^k)^{-1} A^T d_x^k \overset{\text{l.s.}}{=} -D^k g^k, \\ d^k = A^T d_x^k. \end{cases} \tag{3.15}$$

Once $d^k = A^T d_x^k$ is computed, λ can be updated by

$$\lambda^{k+1} \leftarrow (D_r^k)^{-1} D_\theta^k d^k + g^k. \tag{3.16}$$

The new method is referred to as GNCS: a globalized Newton method that uses the complementary slackness conditions for l_p-norm problems. It is summarized in Fig. 4.

Given an initial point $r^0 = A^T x^0 - b$ *with* $|r^0| > 0$ *and* λ^0

Step 1 Compute θ^k by (3.12) and $g^k = p(|r^k|)^{p-1}\sigma^k$; Let $D_r^k = \text{diag}(|r^k|)$, let $D_\theta^k = \text{diag}(|pg^k - (e - \theta^k).*\lambda^k|)$, and define $D^k = (D_r^k (D_\theta^k)^{-1})^{1/2}$;

Step 2 Compute the direction d^k by

$$\begin{cases} (D^k)^{-1} A^T d_x^k \overset{\text{l.s.}}{=} -D^k g^k, \\ d^k = A^T d_x^k; \end{cases}$$

Update λ^{k+1}:

$$\lambda^{k+1} \leftarrow (D_r^k)^{-1} D_\theta^k d^k + g^k;$$

Step 3 Compute τ^k by (3.14); Apply the line search procedure as described in Fig. 3; Update:

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

Go to Step 1;

FIG. 4. *The* GNCS *Algorithm.*

Remark. It is interesting that we can express the fact that the function is smooth through (3.3): optimality conditions simply require $\lambda^* = g^*$. Thus if we ignore the requirement that $\lambda = Z^T w$, we may set

$$\lambda^{k+1} \leftarrow g^{k+1}. \tag{3.17}$$

If this definition of λ^{k+1} is used, $\eta^{k+1} = 0$ and $\theta^{k+1} = 0$. When $\theta^k = 0$, step (3.15) is equivalent to the Newton step (3.6). Hence GNCS becomes IRLSL if we set $\theta^k = 0$ at each iteration. Indeed, GNCS and IRLSL are computationally very similar. The only difference is that for IRLSL, $D_\theta^k = \text{diag}(|(p-1)g^k|)$ and the multiplier information $\{\lambda^k\}$ is not used in defining descent directions. The multipliers are used in GNCS and can be obtained at almost no cost.

In the next section we prove that GNCS is globally convergent for all $1 < p < 2$. Moreover, when there is no $r_i^* = 0$ at a solution we have $(A(D_r^k)^{-1} D_\theta^k A^T) \to (A(D_r^k)^{-1} D_\lambda^k A^T)$ fast enough so that superlinear convergence is achieved.

4. Convergence properties. As we have mentioned before, when $p = 1$, the GNCS algorithm is equivalent to the method Coleman and Li proposed and analyzed in [5]. Thus when $p = 1$ GNCS is globally and quadratically convergent under some nondegeneracy assumptions. We need only to consider the convergence of GNCS when $1 < p < 2$. For the rest of this section we assume that $1 < p < 2$.

For IRLS (without the line search) global convergence is established in [13]. However, the convergence of IRLSL (with the line search) still needs to be established. The convergence for GNCS when $1 < p < 2$ does not follow automatically from the convergence theory [7] for general line-search-based algorithms because the objective function is not twice differentiable everywhere. In addition, our line search procedure is not standard.

We first consider global convergence for both IRLSL and GNCS.

Let P^k be the orthogonal projector onto the orthogonal space of ZD^k, i.e.,

$$P^k = I - D^k Z^T (Z(D^k)^2 Z^T)^{-1} Z D^k.$$

Assume D^k equals either $(D_r^k(\mathrm{diag}((p-1)|g^k|)^{-1})^{1/2}$ or $(D_r^k D_\theta^{k^{-1}})^{1/2}$, depending on whether IRLSL or GNCS is being considered. Then

$$\begin{aligned} d^k &= -A^T(A(D^k)^{-2}A^T)^{-1}Ag^k \\ &= -D^k P^k D^k g^k \\ &= -(D^k)^2(g^k - \lambda^{k+1}), \end{aligned} \tag{4.1}$$

where $\lambda^{k+1} = Z^T w^{k+1}$ and w^{k+1} is the least-squares solution to

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

First, we prove that $\{d^k\}$ generated by each algorithm converges to zero.

THEOREM 4.1. *Let D^k and d^k be defined by* GNCS (*or by* IRLSL). *Then* $\lim_{k\to\infty} \|P^k D^k g^k\|_2 = 0$ *and* $\lim_{k\to\infty} d^k = 0$.

Proof. It is clear that

$$\phi(r^k) - \phi(r^0) = \sum_{j=0}^{k-1} (\phi(r^{j+1}) - \phi(r^j)) \tag{4.2}$$

$$\leq \sum_{j=0}^{k-1} \beta_f \alpha^j g^{j^T} d^j \qquad \text{(from Lemma 2.1)}, \tag{4.3}$$

with $\beta_f > 0$. Since $\{\phi(r^k)\}$ is bounded and $\alpha^k g^{k^T} d^k < 0$ always,

$$\lim_{k\to\infty} \alpha^k g^{k^T} d^k = 0.$$

From the line search procedure we have $\alpha^k \geq \check{\alpha}^k$. Using Lemma 3.2, we have $\alpha^k \geq (p-1)$. Hence

$$\lim_{k\to\infty} g^{k^T} d^k = 0.$$

But $g^{k^T} d^k = -\|P^k D^k g^k\|_2^2$ according to (4.1). This means

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

Since $\phi(r^k)$ is bounded below, $\{r^k\}$ is bounded. From Lemma 3.1 there exists $M > 0$ such that

$$|D^k| \leq \frac{1}{p(p-1)}(\text{diag}(|r^k|^{2-p}))^{1/2} \leq M.$$

Using (4.1) again, we obtain

$$\lim_{k\to\infty} d^k = 0. \qquad \square$$

Next we prove that $\{r^k\}$ converges.

THEOREM 4.2. *Let* $\{r^k\}$ *be obtained by* GNCS (*or by* IRLSL). *Then* $\{r^k\}$ *converges to* r^*.

Proof. Let $\mathcal{S} = \{\bar{r} \ : \ \bar{r} \text{ is a limit point of } \{r^k\}\}$. From Lemma (3.2)

$$\check{\alpha}^k \leq p + 1.$$

With our line search, $\alpha^k \leq \max\{\rho_B, 1, \check{\alpha}^k\}$. Hence $\{\alpha^k\}$ is bounded. From Theorem 4.1 we have

$$\lim_{k\to\infty} \alpha^k d^k = 0.$$

Since $\{r^k\}$ is bounded and $\{\alpha^k d^k\}$ converges to zero, $\mathcal{S}$ is closed and connected [12, p. 478].

Since $\{\phi(r^k)\}$ is monotonically decreasing and bounded below and $\phi(r)$ is continuous, there exists an r^* such that

$$\lim_{k\to\infty} \phi(r^k) = \phi(r^*).$$

Hence for any limit point $\bar{r} \in \mathcal{S}$, $\phi(\bar{r}) = \phi(r^*)$. In addition, since $\mathcal{S}$ is closed and connected and $\phi(r)$ is strictly convex, $\mathcal{S}$ can contain only one point. From the boundedness of $\{r^k\}$ and the uniqueness of its limit point, we conclude that $\{r^k\}$ converges to r^*. $\square$

Finally, we prove that by assuming $r^k = A^T x^k - b$, $\{x^k\}$ converges to a solution of (1.1).

THEOREM 4.3. *Let* $\{r^k\}$ *be obtained by* GNCS (*or by* IRLSL), *and let* $r^k = A^T x^k - b^k$. *Assume that at the limit point* $r^* = A^T x^* - b$, $\{a_i \ : \ b_i - a_i^T x^* = 0\}$ *is a linearly independent set. Then* $\{\lambda^k\}$ *converges to* λ^* *and* $\{x^k\}$ *converges to the solution of* (1.1).

Proof. Following Theorem 4.2 there exists r^* such that $\lim_{k\to\infty} r^k = r^*$. Thus there exists x^* with $\lim_{k\to\infty} x^k = x^*$.

Let $Z = [z_1, \ldots, z_m]$, and let $\mathcal{A}_c^* = \{i \mid r_i^* \neq 0\}$. Since $\lim_{k\to\infty} D^k(g^k - Z^T w^{k+1}) = 0$, any limit point $\bar{w}$ of $\{w^{k+1}\}$ satisfies $z_i^T \bar{w} = g_i^*$, $\forall i \in \mathcal{A}_c^*$. By assumption that at the limit point r^*, $\{a_i \ : \ b_i - a_i^T x^* = 0\}$ is linearly independent, $z_i^T w = g_i^*, i \in \mathcal{A}_c^*$ has a unique solution. Hence $\{\lambda^k = Z^T w^k\}$ is bounded and converges to λ^*.

We prove that x^* is a solution by showing that $\lambda_i^* = 0$ if $r_i^* = 0$. Assume otherwise, i.e., that there exists some $\lambda_j^* \neq 0$ with $r_j^* = 0$. Consider the breakpoint α_j^k as defined by (2.10). Then

$$\alpha_j^k = \begin{cases} \dfrac{\sigma_j |p g_j^k - (1-\theta_j^k)\lambda_j^k|}{g_j^k - \lambda_j^{k+1}} & \text{for GNCS,} \\ \dfrac{r_j^k}{(|r_j^k|)^{2-p}(g_j^k - \lambda_j^{k+1})} & \text{for IRLSL.} \end{cases}$$

It is clear that $\{\alpha_j^k\}$ converges to zero because $\{g_j^k\}$ and $\{1-\theta_j^k\}$ converge to zero. Hence there exists k_1 such that when $k \geq k_1$, $\alpha_j^k < \check{\alpha} = p-1$, all nonzero λ_j^k remain the same sign and $|r_j^k|^{p-1} < |\lambda_j^k|$ for all $r_j^* = 0$ with $\lambda_j^* \neq 0$.

By using Lemma 3.2 (or Lemma 2.1), $\alpha^k > \check{\alpha}^k > \alpha_j^k$ for $k \geq k_1$. If $r_j^{k_1}$ and $\lambda_j^{k_1+1}$ have different signs, at iteration $k = k_1 + 1$, λ_j^{k+2} and r_j^{k+1} will have the same sign because $\alpha^k > \alpha_j^k$. If, for $\hat{k} > k_1$, $r_j^{\hat{k}}$ and $\lambda_j^{\hat{k}+1}$ have the same sign, it will remain so for $k > \hat{k}$ because $g_j^k d_j^k > 0$. But this means $|r_j^k|$ will be increased for $k > k_1$. This contradicts the fact that $r_j^* = 0$. □

Now we discuss the local convergence properties of the two algorithms. If at the solution r^* there is some $r_i^* = 0$, the Hessian matrix of $\psi(x)$ does not exist at a corresponding x^*. Hence, theoretically, we do not expect superlinear convergence for either the IRLSL or the GNCS algorithm.

Assume that $r_i^* \neq 0$ for any $1 \leq i \leq m$. The Hessian matrix of $\psi(x)$ is positive definite at x^*. Following Theorem 4.1, $\{d^k\}$ converges to zero. If our line search procedure is used, every positive breakpoint α_i^k converges to infinity. This means that the unit step size is tested for acceptance for sufficiently large k. Since $\psi(x)$ is twice continuously differentiable near x^*, a unit step size is admissible for a Newton step or a quasi-Newton step (e.g., [7]). In addition, perturbation to the unit step size is not necessary close to the solution because $|r^k + d^k| > 0$ for sufficiently large k. Hence $\alpha^k = 1$ for sufficiently large k. Thus the IRLSL method is locally equivalent to the Newton method for minimizing $\psi(x)$, which is a locally twice continuously differentiable function. Hence, if standard unconstrained minimization convergence analysis is followed (e.g., [7]), the IRLSL method is locally quadratically convergent. Similarly, the GNCS algorithm is locally equivalent to a quasi-Newton method for the minimization of a twice continuously differentiable function $\psi(x)$, with the Hessian matrix replaced by the matrix $A^T(D_r^k)^{-1}D_\theta^k A$. Moreover, we have

$$\lim_{k\to\infty} \frac{\|\nabla\psi(x^k) - \nabla^2\psi(x^k)d_x^k\|}{\|d_x^k\|} = \lim_{k\to\infty} \frac{\|(A^T D_r^{k^{-1}} D_\theta^k A - A^T D_r^{k^{-1}} \mathrm{diag}(|(p-1)g^k|)A)d_x\|}{\|d_x^k\|} = 0$$

since $\{D_\theta^k - \mathrm{diag}((p-1)|g^k|)\}$ converges to zero. From [6, Thm. 6.4], $\{x^k\}$ converges superlinearly to x^*.

In summary, we have shown that under the assumptions of Theorem 4.3 a sequence $\{r^k\}$ generated by either IRLSL or GNCS from any starting point $r^0 = A^T x^0 - b$ with $|r^k| > 0$ converges to a solution. If it is assumed that there is no zero residual at the solution, IRLSL is *locally quadratically* convergent, whereas the GNCS method is *locally superlinearly* convergent.

5. Numerical experiments. In this section we compare the computational performance of the IRLSL method with that of the proposed GNCS algorithm. All the experiments are done in MATLAB [11]. The numerical results clearly show the superiority of GNCS over IRLSL (and thus over IRLS as well).

The dominant cost of the computation of the two methods is the same: solving a weighted least-squares problem of the same dimension and structure per iteration. Moreover, the same line search procedure is used.

Now we discuss possible stopping criteria for problem (1.1).

Assume $1 < p < 2$. The optimality condition is simply $Ag^* = 0$ or, equivalently, $\eta^* = 0$ (see (3.12)). We point out, however, that testing $\|Ag\|$ against a tolerance is generally not a good stopping criterion. When p is close to unity, the gradient function $A(D_r)^{p-1}\sigma$ is ill conditioned in the neighborhood of a point where some $r_i = 0$, i.e., a small change of a variable may lead to a large change in the gradient. As an example, let us consider a simple scalar function $\psi(\xi) = |\xi|^{1.001}$. The gradient function is equal to $1.001\xi^{0.001}\,\mathrm{sgn}(\xi)$. Even when $\xi = 2.2204 \times 10^{-16}$ (machine precision in MATLAB), the gradient $\nabla\psi(\xi)$ equals 0.9656. Since the gradient should be zero when $\xi = 0$, it is clear that the gradient function is extremely unstable.

In our computation we terminate the calculation when the algorithm has stopped decreasing the objective function. More specifically, we stop the computation when

$$\textbf{either } \frac{|\phi(r^{k+1}) - \phi(r^k)|}{\phi(r^{k+1})} < \tau_s \textbf{ or } \eta^k < \tau_s \textbf{ or } \text{itcount} > 50,$$

where τ_s has been set to $\frac{1}{2}10^{-11}$ and itcount denotes the number of iterations. For the GNCS algorithm, if $p = 1$ or if $1 < p < 2$ but there is no zero residual at the solution, we observe that final superlinear or quadratic convergence is achieved and that the accuracy of the computed solution is about τ_s (since η^k is about τ_s at termination). For the IRLSL method this is true only when $1 < p < 2$ and there is no zero residual at the solution.

For the results reported in this paper the parameters required by the algorithms are set as follows:

$$\tau \leftarrow 0.975, \qquad \beta_f \leftarrow \epsilon, \qquad \gamma \leftarrow 0.99,$$

where ϵ is machine precision.

TABLE 3
Function approximation problems.

$m = 200$, $n = 6$, $f_1(z)$		
p	GNCS	IRLSL
1	11	20
1.001	13	30
1.01	12	25
1.1	11	22
1.2	10	45
1.3	8	26
1.4	9	22
1.5	8	17
1.6	7	17
1.7	6	12
1.8	5	8
1.9	4	5

$m = 200$, $n = 10$, $f_2(z)$		
p	GNCS	IRLSL
1	12	50
1.001	11	50
1.01	15	50
1.1	10	33
1.2	9	23
1.3	7	27
1.4	8	20
1.5	6	15
1.6	6	12
1.7	6	11
1.8	6	7
1.9	4	6

The starting point for both IRLSL and GNCS is computed as the solution to $A^T x \overset{\text{l.s.}}{=} b$. Our experience indicates that the role of λ^0 is less significant, and we set it in a similar way to that defined in [5]:

$$\lambda^0 = \tau \frac{g^0}{\max(|r^0|)}.$$

Next, we generate some test problems from discrete approximation.

Function approximation problems. Approximate $f(z)$, evaluated at $z = 0, \frac{1}{m}, \ldots, 1$, by a polynomial of degree $n-1$: $\sum_{j=1}^{n} x_j z^{j-1}$ such that the l_p-norm residuals are minimized. The two test functions used are

$$f_1(z) = \sqrt{1+z}, \qquad f_2(z) = e^z + \begin{cases} 5 & \text{if } 0.1 < z < 0.2, \\ 0 & \text{otherwise.} \end{cases}$$

As indicated by Table 3, GNCS is consistently better than IRLSL. The first function $f_1(z)$ is continuous, whereas the second function $f_2(z)$ is not. For $f_1(z)$ and $p = 1.9$, the best l_p-norm residual is $\phi(r^*) = 4.97528518113 \times 10^{-10}$. For the second function $f_2(z)$, if $p = 1.9$, the best l_p-norm residual is $\phi(r^*) = 1.7535105 \times 10^2$.

Random problems. We also generate random test problems by generating random entries for matrix A and right-hand side b by using the random number generator (with normal distribution) in PRO-MATLAB [11].

TABLE 4
$p = 1$.

Number of Steps	$m = 100$	
n	GNCS	IRLSL
10	12	50
30	14	50
50	12	50
70	13	50
90	14	50

Number of Steps	$m = 200$	
n	GNCS	IRLSL
10	17	50
30	17	50
50	15	50
70	21	50
90	15	50
110	14	50
130	17	50
150	13	50
170	13	50
190	9	50

TABLE 5
$p = 1.001$.

Number of Steps	$m = 100$	
n	GNCS	IRLSL
10	11	27
20	14	46
30	20	50
40	16	50
50	16	50
60	17	50
70	14	50
80	11	50
90	13	37

Number of Steps	$m = 200$	
n	GNCS	IRLSL
10	15	38
30	18	50
50	15	50
70	17	50
90	21	50
110	15	50
130	17	50
150	14	50
170	18	50
190	13	50

Table 4 exhibits the number of iterations required by GNCS and IRLSL when $p = 1$. The IRLSL method stops after 50 iterations with the objective function having only a few digits of accuracy. The GNCS algorithm is essentially the method presented in [5], and it demonstrates fast convergence.

When p is very close to unity (e.g., see Tables 5 and 6 with $p = 1.001, 1.01$), the number of zero residuals at a solution is usually slightly less than n. The GNCS algorithm exhibits final superlinear convergence behavior because n of the residuals

are usually nearly zero at the solution. The GNCS algorithm behaves as though approaching a vertex and thus demonstrates superlinear behavior when approaching the neighborhood of the solution. At termination the objective function values computed by GNCS are always smaller than that of IRLSL. Comparing the IRLSL solutions with the more accurate GNCS solutions, we see that the former typically have about six digits of accuracy. The IRLSL method again shows extremely slow convergence and fails to find a solution after 50 iterations for the majority of problems.

TABLE 6
$p = 1.01$.

Number of Steps $m = 100$		
n	GNCS	IRLSL
10	12	34
30	12	50
50	13	50
70	13	50
90	16	50

Number of Steps $m = 200$		
n	GNCS	IRLSL
10	11	33
30	18	50
50	18	48
70	19	41
90	17	50
110	17	50
130	17	50
150	15	47
170	13	50
190	17	50

When p is further away from unity (e.g., Tables 7 and 8 with $p = 1.1, 1.3$), the number of zero residuals at the solution is less. However, many residuals are still relatively small. Hence the GNCS algorithm again approaches the neighborhood of a solution with a few final superlinear steps. Here the IRLSL method finds a solution with the required accuracy, but the number of iterations required by IRLSL is more than twice of that of the GNCS algorithm (see Tables 7 and 8).

When p is significantly larger than unity (e.g., Table 9 with $p = 1.7$), there usually exists no zero residual at the solution. Thus both the GNCS algorithm and the IRLSL method converge quickly to solutions and exhibit fast convergence. For these problems the two methods have roughly the same behavior.

In summary, the GNCS algorithm works very well for all $1 \leq p < 2$. It always performs significantly better than IRLSL when p is close to unity ($p < 1.5$). When p is *very close or equal* to unity, the IRLSL method is extremely inefficient, whereas the GNCS method finds the solutions in about 18 iterations. The latter is slightly better than IRLSL when $p \geq 1.5$ and there exists no zero residual at a solution.

Finally, we point out that the number of iterations required by the GNCS method appears to be relatively insensitive to the problem size.

6. Conclusions. In this paper we have developed a new efficient method that solves the l_p-norm minimization problem with $1 \leq p < 2$. We also have further investigated the performance of the classical IRLS method and have compared it with the new approach. We observed that the slow convergence of the IRLS (or IRLSL) method is not entirely due to the zero residuals at a solution but is also due to the fact that the constrained aspect is not taken care of: the Newton steps for the IRLS methods are based on the optimality conditions for the unconstrained problem ($1 < p < 2$) but not the constrained case ($p = 1$). On the basis of this observation we developed the GNCS method, which uses the Newton directions derived from the optimality conditions for all $1 \leq p < 2$.

TABLE 7
$p = 1.1$.

Number of Steps	$m = 100$	
n	GNCS	IRLSL
10	11	19
30	9	24
50	11	24
70	10	25
90	10	28

Number of Steps	$m = 200$	
n	GNCS	IRLSL
10	10	15
30	11	26
50	12	28
70	11	26
90	11	29
110	10	34
130	10	27
150	12	27
170	10	37
190	10	32

TABLE 8
$p = 1.3$.

Number of Steps	$m = 100$	
n	GNCS	IRLSL
10	7	8
30	8	10
50	8	11
70	9	13
90	8	15

Number of Steps	$m = 200$	
n	GNCS	IRLSL
10	8	13
30	9	13
50	8	13
70	8	13
90	8	13
110	9	15
130	9	16
150	9	17
170	9	17
190	9	19

TABLE 9
$p = 1.7$.

Number of Steps	$m = 100$	
n	GNCS	IRLSL
10	6	5
30	6	8
50	7	7
70	9	7
90	8	11

Number of Steps	$m = 200$	
n	GNCS	IRLSL
10	5	5
30	6	7
50	6	7
70	6	6
90	7	7
110	7	7
130	6	6
150	7	10
170	8	9
190	7	11

The GNCS method is attractive because of its capability to efficiently solve the l_p-norm minimization problem with the entire range $1 \leq p < 2$. When $p = 1$ it is exactly the approach for l_1 presented in [5] and is quadratically convergent under nondegeneracy assumptions. When $p > 1$ the new method is superlinearly convergent when there are no zero residuals at the solution.

The GNCS method is significantly better than the IRLSL algorithm when p is close or equal to unity. The computational cost of each iteration of the two methods is the same: the main cost is solving a weighted least-squares problem of the same

size and structure. The difference between the two methods lies only in the definition of the different diagonal scaling matrices that define descent directions: in our new method the multiplier information is incorporated in the diagonal scaling matrix, and this is the key to a significant improvement.

Acknowledgments. The author thanks Thomas Coleman for numerous helpful suggestions. The author is also grateful to Mike Overton and the anonymous referees, whose suggestions improved the presentation of the paper.

REFERENCES

[1] I. BARRODALE AND F. ROBERTS, *An improved algorithm for discrete l_1 linear approximation*, SIAM J. Numer. Anal., 10 (1973), pp. 839–848.

[2] R. H. BARTELS, A. R. CONN, AND J. W. SINCLAIR, *Minimization techniques for piecewise differentiable functions: The l_1 solution to an overdetermined linear system*, SIAM J. Numer. Anal., 15 (1978), pp. 224–240.

[3] R. H. BYRD, *Algorithms for robust regression*, in Nonlinear Optimization 1981, M. Powell, ed., Academic Press, New York, 1982, pp. 79–89.

[4] T. F. COLEMAN AND Y. LI, *A global and quadratically-convergent method for linear l_∞ problems*, SIAM J. Sci. Statist. Comput., 29 (1992), pp. 1166–1186.

[5] ———, *A globally and quadratically convergent affine scaling method for linear l_1 problems*, Math. Programming, 56 (1992), pp. 189–222.

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

[7] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[8] J. FISHER, *An algorithm for discrete linear l_p approximation*, Numer. Math., 38 (1981), pp. 129–139.

[9] P. GILL, W. MURRAY, AND M. WRIGHT, *Practical Optimization*, Academic Press, New York, 1981.

[10] G. MERLE AND H. SPÄTH, *Computational experience with discrete l_p-approximation*, Computing, 12 (1974), pp. 315–321.

[11] C. B. MOLER, J. LITTLE, S. BANGERT, AND S. KLEIMAN, *ProMatlab User's Guide*, MathWorks, Sherborn, MA, 1987.

[12] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

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

[14] G. A. WATSON, *On two methods for discrete l_p approximation*, Computing, 18 (1977), pp. 263–266.

[15] J. M. WOLFE, *On the convergence of an algorithm for discrete l_p approximation*, Numer. Math., 32 (1979), pp. 439–459.