SIAM J. OPTIMIZATION
Vol. 3, No. 2, pp. 298–321, May 1993
© 1993 Society for Industrial and Applied Mathematics
005

A GLOBALLY AND SUPERLINEARLY CONVERGENT ALGORITHM FOR CONVEX QUADRATIC PROGRAMS WITH SIMPLE BOUNDS*

THOMAS F. COLEMAN† AND LAURIE A. HULBERT‡

Abstract. A globally and superlinearly convergent algorithm for solving convex quadratic programs with simple bounds is presented. The algorithm is developed using a new formulation of the problem: the minimization of an unconstrained piecewise quadratic function that has the same optimality conditions as the original problem. The major work at each iteration is the Cholesky factorization of a positive definite matrix with the size and structure of the Hessian of the quadratic. Hence, the algorithm is suitable for solving large sparse problems and for implementation on parallel computers. The numerical results indicate that the new approach has promise.

Key words. quadratic programming, interior point methods, simple bounds, box constraints, large sparse minimization

AMS subject classifications. 90C20, 65K05

1. Introduction.

In this paper, we present a new algorithm for solving the problem

$$\min \tfrac{1}{2}x^T Ax + b^T x \qquad -1 \le x \le 1, \tag{1}$$

where A is an $n \times n$ symmetric positive definite matrix. In theory our approach can be applied to problems with general upper and lower bounds after a simple transformation to yield form (1). In practice this works without difficulty provided the ranges are not extreme. When there are large ranges, numerical difficulties may prevent an accurate solution. However, we believe that in many practical instances it is often the case that reasonable feasibility ranges are known in advance.

Many algorithms, both finite and infinite, have been proposed for (1). *Finite* algorithms (assuming exact arithmetic), usually involving pivoting and determination of an "active-set," are the most common. Recent contributions include: Björck [1], Coleman and Hulbert [3], Dembo and Tulowitzki [5], Júdice and Pires [9], Lötstedt [11], Moré and Toraldo [12], Öreborn [14], O'Leary [13], and Yang and Tolle [17].

Following Karmarkar's [10] development of an (infinite) "interior-point" algorithm for linear programming, there has been increased interest in infinite interior-point algorithms for quadratic programs. Interior-point algorithms for quadratic programs are typically based on affine scaling, path following or barrier functions, potential reduction, or projection techniques and are in general simpler to implement than active-set methods because they require less data structure manipulation. For a discussion of recent interior-point algorithms for this and other quadratic programs, see the survey paper by Ye [19]. Some of these interior-point algorithms have polynomial time bounds,[1] but

*Received by the editors February 10, 1990; accepted for publication (in revised form) January 22, 1992. 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 and by the Computational Mathematics Program of the National Science Foundation under grant DMS-8706133.

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

‡Department of Mathematics and Computer Science, James Madison University, Harrisonburg, Virginia 22807.

[1] One can also take a finite view of such algorithms, assuming integer data, exact arithmetic, and a formal final "rounding" to the exact solution. This view leads to a complexity analysis; e.g., is the number of steps bounded by a polynomial in the size of the problem? This is not our concern in this paper.

their asymptotic rates of convergence have not been studied. The affine scaling method proposed by Ye [18] which is similar to but simpler than the polynomial algorithm of Ye and Tse [20] and has no proven polynomial time bound, displays linear convergence in practice. While few numerical results are available for the recent polynomial algorithms, those presented by Han, Pardalos, and Ye [8] show that the performance of their polynomial algorithm is more consistent than that of active-set algorithms.

We present a new infinite algorithm here that is not an interior-point method. In general, infeasible iterates are generated. Our algorithm is globally and superlinearly convergent; however, we do not claim that it has a polynomial time bound. We develop our algorithm using a new formulation of the problem: the minimization of an unconstrained piecewise quadratic function that has the same optimality conditions as the original problem. Our algorithm has similarities to an l_1 penalty function method, and is quite similar in development to the quadratically convergent affine scaling method for the linear l_1 problems of Coleman and Li [4]. The major work at each iteration of our algorithm is the Cholesky factorization of a positive definite matrix with the size and structure of the matrix A. Hence our algorithm is suitable for solving large sparse problems and for implementation on parallel computers.

There are three basic ideas underlying our new approach. The major purpose of this paper is to expose these ideas and to begin to explore their potential in constrained optimization. The first idea is the observation, detailed below, that a simple transformation changes (1) into an unconstrained minimization problem involving a piecewise quadratic function $f(y)$. This allows for the possibility of using unconstrained minimization strategies. The second idea, discussed in §2, is that there is a well-defined unconstrained Newton process in a neighborhood of the solution. This Newton process is defined with respect to the optimality conditions. The third idea is the definition of a descent direction, and a piecewise line search procedure that ultimately leads to full Newton steps, thereby ensuring superlinear convergence.

The paper is organized as follows. In the rest of this section, we present our new formulation of the problem and introduce some notation and definitions. In §2, we describe and motivate our algorithm. We prove global convergence in §3 and superlinear convergence in §4. Section 5 contains numerical results and a discussion of the behavior of the algorithm. Finally, in §6, we discuss possible improvements and make some concluding remarks.

1.1. A related problem. Consider the quadratic program (1). Let $q_x(x) = \frac{1}{2}x^T Ax + b^T x$, and hence $\nabla q_x(x) = Ax + b$. If we assume that x^* satisfies $\nabla q_x(x^*)_i \neq 0$ for all i such that $|x_i^*| = 1$, then the following conditions are sufficient to guarantee that x^* is a local minimum of (1):

$$\text{feasibility:} \quad -1 \leq x^* \leq 1,$$

$$\text{first order:} \quad \begin{cases} \nabla q_x(x^*)_i = 0 & \text{if } -1 < x_i^* < 1, \\ \nabla q_x(x^*)_i < 0 & \text{if } x_i^* = 1, \\ \nabla q_x(x^*)_i > 0 & \text{if } x_i^* = -1. \end{cases}$$

Now for a vector v, define the vector-valued function sign(v), where

$$\text{sign}(v)_i = \begin{cases} 1 & \text{if } v_i \geq 0, \\ -1 & \text{if } v_i < 0. \end{cases}$$

Then letting $d_i = x_i^* + \text{sign}(\nabla q_x(x^*))_i$, and $D = \text{diag}(d_i)$, we can express the first-order condition as

$$D\nabla q_x(x^*) = 0. \tag{2}$$

Now consider the following piecewise quadratic minimization problem:

$$\begin{aligned} \min f(y) &= \tfrac{1}{2} y^T A^{-1} y + y^T A^{-1} b + \|y\|_1 \\ &= q_y(y) + \|y\|_1, \end{aligned} \tag{3}$$

where $q_y(y) = \frac{1}{2} y^T A^{-1} y + y^T A^{-1} b$. The following conditions are sufficient to guarantee that y^* is a minimum of (3) [2]: there exists a vector λ^* such that

$$A^{-1}y^* + A^{-1}b + \sum_{i \ni y_i^* \neq 0} \text{sign}(y_i^*) e_i = - \sum_{i \ni y_i^* = 0} \lambda_i^* e_i, \qquad -1 \leq \lambda^* \leq 1.$$

We can reformulate these conditions into the following equivalent conditions: there exists a vector λ^* such that

$$\begin{aligned} &Y^*(-\lambda^* + \text{sign}(y^*)) = 0, \\ &\lambda^* = -(A^{-1}y^* + A^{-1}b), \qquad -1 \leq \lambda^* \leq 1, \end{aligned} \tag{4}$$

where $Y^* = \text{diag}(y^*)$. Thus if we equate λ^* with x^* and hence y^* with $-\nabla q_x(x^*)$, then it is apparent that (4) is equivalent to (2) plus feasibility.

This new formulation gives us a new perspective from which to approach solving (1) and this is the view we take in this paper.

For convenience in what follows, we sometimes switch between the original variables x and $-\nabla q_x(x)$ and the new variables λ and y:

$$x = \lambda, \qquad y = -\nabla q_x(x). \tag{5}$$

In general, we develop our algorithm and prove things about it in the λ and y variables and describe the characteristics of the quadratic programs in the x and $\nabla q_x(x)$ variables.

1.2. Some notation and definitions. In what follows, subscripts denote vector and matrix components and superscripts denote iteration number. We omit superscripts whenever the iteration number is clear or irrelevant. For any vector v, the matrix $\text{diag}(v)$ is a diagonal matrix whose diagonal elements are the components of v. If V is a matrix, let $|V|$ be the matrix whose ijth element is $|v_{ij}|$.

For any point y with $y_i \neq 0$ for all i, $\nabla f(y)$ is defined and $\nabla f(y) = A^{-1}y + A^{-1}b + \text{sign}(y)$. Given y and s, define a *breakpoint* of f along s to be any α where $f(y + \alpha s)$ is nondifferentiable, i.e., $(y + \alpha s)_i = 0$ for some i. For $\alpha > 0$ define $S(\alpha, y, s)$ to be the set of indices to breakpoints along s that occur at or before α, i.e.,

$$S(\alpha, y, s) = \{i \mid 0 < -y_i/s_i \leq \alpha\}. \tag{6}$$

Define $\sigma(\alpha, y, s) = \text{sign}(y + \alpha s)$. For any direction s, define

$$g(\alpha, y, s) = \lim_{\eta \to \alpha^+} \nabla f(y + \eta s).$$

Notice that if $(y + \alpha s)_i \neq 0$ for all i, then $g(\alpha, y, s) = \nabla f(y + \alpha s)$. For conciseness, we write $g(\alpha)$, $S(\alpha)$, and $\sigma(\alpha)$ when y and s are clear from context, and, in particular, $g^k(\alpha)$, $S^k(\alpha)$, and $\sigma^k(\alpha)$ when $y = y^k$ and $s = s^k$. Also, since we use it so frequently, we let σ denote $\sigma(0)$, i.e., $\sigma = \text{sign}(y)$.

A point satisfying $|x_i| = 1$ and $\nabla q_x(x)_i = 0$ for some i is called a *degenerate* point. We call a quadratic program of the form in (1) *nondegenerate on a closed bounded set* C if at every point $x \in C$ either $|x_i| \neq 1$ or $\nabla q_x(x)_i \neq 0$.

The nondegeneracy assumption. Given a closed bounded set C, the nondegeneracy assumption, with respect to C, is that at every point $x \in C$ either $|x_i| \neq 1$ or $\nabla q_x(x)_i \neq 0$.

2. The algorithm. Problem (3) is an unconstrained optimization problem; therefore, a descent direction algorithm can be developed without regard to maintaining feasibility. On the other hand, $f(y)$ is not everywhere differentiable due to the l_1-term $\|y\|_1$. The challenge is to deal with this piecewise nature of f. In response, our algorithm restricts iterates to differentiable points; i.e., $y_i^k \neq 0$ for all iterations k and components i.

2.1. The search direction. From (4), we see that a solution[2] to (3) is also a zero of

$$F(y) = Y(A^{-1}y + A^{-1}b + \text{sign}(y)) = 0. \tag{7}$$

Although F is not differentiable whenever $y_i = 0$ for some i, at all other points $F(y) = Y\nabla f(y)$ and is twice continuously differentiable. This naturally suggests using Newton's method, at least in a neighborhood of y^*. Where it is defined, the Jacobian of $F(y)$ is

$$J(y) = YA^{-1} + \text{diag}(\nabla f(y)),$$

and thus the Newton step for F at y is

$$s_N = -(YA^{-1} + \text{diag}(\nabla f(y)))^{-1} Y \nabla f(y). \tag{8}$$

The following lemma shows that in a neighborhood of the solution of (3), the Newton step for F is a descent direction for $f(y)$. This is not an obvious result since the Newton process does not come directly from f but from the nonlinear system of equations (7). The idea behind the proof is that $\nabla f(y)_i / y_i$ either converges to zero or to (positive) infinity as $y \to y^*$. Specifically, if $y_i^* \neq 0$, then $\nabla f(y)_i / y_i$ converges to 0; if $y_i^* = 0$, then $\nabla f(y)_i / y_i$ converges to $+\infty$. Consequently, the matrix $(A^{-1} + \text{diag}(\nabla f(x))/Y)$ is positive definite in a neighborhood of y^*; therefore, by (8), s_N will be a descent direction.

LEMMA 2.1. *Assume nondegeneracy of* (1) *at the solution. Then, there exists* $\epsilon > 0$ *such that whenever* $y_i \neq 0$ *for all* i *and* $\|y - y^*\| < \epsilon$, *we have* $-s_N^T \nabla f(y) > 0$.

Proof. Rewriting (8) as

$$s_N = -(A^{-1} + Y^{-1}\text{diag}(\nabla f(y)))^{-1} \nabla f(y), \tag{9}$$

we can see that if $\nabla f(y)_i / y_i > -(1/\|A\|_2)$ for all i, where $(1/\|A\|_2)$ is the smallest eigenvalue of A^{-1}, then $-s_N^T \nabla f(y) > 0$. Set

$$\epsilon = \frac{1}{2} \min \left(\frac{xdg}{\|A^{-1}\|_2}, \frac{ydg}{\|A\|_2 \|A^{-1}\|_2} \right),$$

[2]Recall: We have assumed A is symmetric positive definite, so A^{-1} exists.

where

$$ydg = \min_{\{i:y_i^* \neq 0\}} |y_i^*| \quad \text{and} \quad xdg = \min_{\{i:y_i^*=0\}} (1 - |x_i^*|).$$

Assume $\|y - y^*\|_2 < \epsilon$ and $y_i \neq 0$ for all i.

If $y_i^* \neq 0$ then $\text{sign}(y^*)_i = \text{sign}(y)_i$ by our choice of ϵ, so

$$\begin{aligned} |\nabla f(y)_i| &= |\nabla f(y)_i - \nabla f(y^*)_i| \\ &= |x_i - x_i^*| \\ &\leq \|x - x^*\|_2 \\ &\leq \|A^{-1}\|_2 \|y^* - y\|_2 \\ &< \|A^{-1}\|_2 \, \epsilon \\ &\leq ydg/(2\|A\|_2). \end{aligned}$$

Since

$$|y_i| \geq ydg - \epsilon \geq ydg/2,$$

we have

$$\left| \frac{\nabla f(y)_i}{y_i} \right| < \frac{1}{\|A\|_2}.$$

Hence $\nabla f(y)_i / y_i > -(1/\|A\|_2)$ if $y_i^* \neq 0$.

If $y_i^* = 0$ then

$$| \, |x_i| - |x_i^*| \, | \leq |x_i - x_i^*| < \|A^{-1}\|_2 \, \epsilon \leq xdg/2,$$

so

$$|x_i| < |x_i^*| + xdg/2 \leq 1 - xdg + xdg/2 < 1.$$

And since $\nabla f(y)_i = (\sigma_i - x_i)$, we conclude that $\sigma_i = \text{sign}(\nabla f(y))_i$ and hence $\nabla f(y)_i / y_i > 0$. Thus $\nabla f(y)_i / y_i > -(1/\|A\|_2)$ for all i, so we are done. □

Of course, the Newton step may not be a descent direction far from the solution. Therefore, we consider a "modified" Newton step. Specifically, we choose a step of the form

$$s = -(|Y|A^{-1} + R)^{-1} |Y| \nabla f(y), \tag{10}$$

where R is a diagonal matrix satisfying $r_{ii} > 0$ for all i. Thus we have the following lemma.

LEMMA 2.2. *For any diagonal matrix R with positive diagonal entries, the search direction s, defined by* (10), *is a descent direction, i.e.*, $-s^T \nabla f(y) > 0$.

In order for s to approach the Newton step, we choose $R = \text{diag}(r)$, where

$$r_i = \theta + (1 - \theta)|\nabla f(y)_i|, \tag{11}$$

$\theta \geq 0$, and $\theta = 0$ only at the optimal solution y^*. We define η to quantify the nonoptimality of the current point,

$$\eta = \rho \|Y \nabla f(y)\|_1 + \sum_i \max((|\lambda_i| - 1), 0),$$

where $\rho = 1/\|Y\nabla f(y)\|_1$, evaluated at a "typical" value of y. Our choice of a "typical" value of y is $-(A \cdot \text{sign}(-b) + b)$. Then we choose θ to be between 0 and some small constant $c_1 \in (0,1)$ by setting

$$\theta = c_1\eta/(0.99 + \eta).$$

Notice that either $\nabla f(y^*)_i = 0$ or $\text{sign}(\nabla f(y^*))_i = \sigma_i$, so R, as defined by (11), approaches $\Sigma \cdot \text{diag}(\nabla f(y))$, where $\Sigma = \text{diag}(\sigma_i)$, thus ensuring that the Newton step is approached. We can prove the following useful lemma about R.

LEMMA 2.3. *If r is defined by* (11), *then for all i, $r_i = 0 \iff \theta = 0$ and $\nabla f(y)_i = 0$.*

Proof. By definition, $r_i = \theta + (1-\theta)|\nabla f(y)_i|$. Since $0 \le \theta \le 1$, each term is greater than or equal to zero. Thus $r_i = 0$ if and only if $\theta = 0$ and $\nabla f(y)_i = 0$. □

2.2. The line search. The basic iteration in our overall procedure has the form

$$y^{k+1} = y^k + \alpha^k s^k, \tag{12}$$

where α^k is the step length, determined after computing the search direction s^k. Before describing this line search procedure, we introduce some notation and describe the geometry of the line search (we drop the superscript k in this discussion since we are referring to a single iteration of the overall procedure).

Define the function $f_{y,s}(\nu)$ to be the restriction of the function f to the line through y along s, i.e.,

$$f_{y,s}(\nu) = f(y + \nu s).$$

Thus $f_{y,s}(\nu)$ is continuous, convex, and piecewise quadratic. Define β to be the vector of positive values of ν where $f_{y,s}(\nu)$ is nondifferentiable, i.e.,

$$\beta_i = \begin{cases} -y_i/s_i & \text{if } \text{sign}(y_i) = -\text{sign}(s_i), \\ \infty & \text{otherwise.} \end{cases}$$

On the interval between any two adjacent breakpoints, say β_i and β_j, $f_{y,s}(\nu)$ is a quadratic. (The breakpoints β_i and β_j are adjacent, with $\beta_i \le \beta_j$, if there does not exist an index k such that $\beta_i < \beta_k < \beta_j$.) Label this quadratic $f_{(i,j)}(\nu)$. Hence the minimum of $f_{y,s}(\nu)$ occurs either at a breakpoint or at the minimum of one of the quadratic segments $f_{(i,j)}(\nu)$. Furthermore, $f'_{(i,j)}(\nu) = s^T g(\nu, y, s)$ and $f''_{(i,j)}(\nu) = s^T A^{-1} s$. Thus on each interval, the function $f'_{y,s}(\nu)$ is a line with slope $s^T A^{-1} s$, i.e., the curvature of f is the same for all intervals. For any ν, let β_i and β_j be the two adjacent breakpoints surrounding ν (i.e., β_i is the largest breakpoint equal to or to the left of ν, β_j is the smallest breakpoint strictly to the right of ν), and define $\gamma(\nu)$ to be the step from ν to the minimum of $f_{(i,j)}(\nu)$, i.e.,

$$\gamma(\nu) = \frac{-s^T g(\nu, y, s)}{s^T A^{-1} s}.$$

For notational convenience, define $\beta_0 = 0$ and $\gamma_i = \gamma(\beta_i)$. Figures 1 and 2 illustrate these quantities where we assume $\beta_0 < \beta_1 < \beta_2$.

In the next lemma, we show that γ is monotonically decreasing. This implies that as we move along the direction s during the line search, the distance to the optimal point

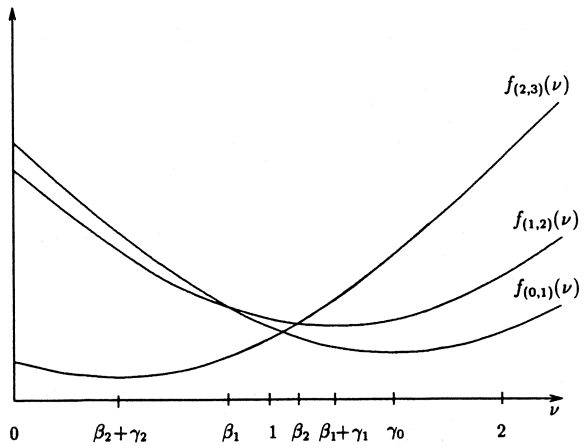

FIG. 1. *The quadratic functions that comprise the piecewise quadratic* $f_{y,s}(\nu)$.

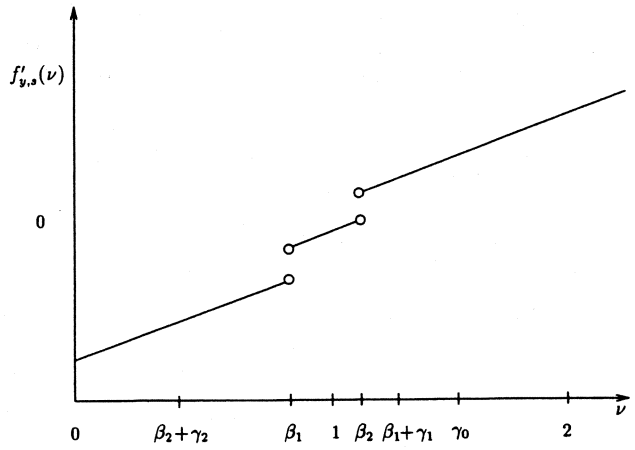

FIG. 2. *The function* $f'_{y,s}(\nu)$.

of the current quadratic is less than the distance to the optimal points of the previously encountered quadratics.

LEMMA 2.4. *Let s be a descent direction for f at the current point. Then, the functions $-s^T g(\nu, y, s)$ and $\gamma(\nu)$ are monotonically decreasing functions of ν.*

Proof. We have

$$-s^T g(\nu, y, s) = -s^T g(0) - \nu s^T A^{-1} s - s^T(\sigma(\nu) - \sigma(0)). \tag{13}$$

But

$$\sigma(\nu)_i - \sigma(0)_i = \begin{cases} 0 & \text{if } \nu < \beta_i, \\ -2\sigma(0)_i & \text{otherwise.} \end{cases}$$

So

$$s^T(\sigma(\nu) - \sigma(0)) = \sum_{i \in S(\nu)} (-2\sigma_i s_i) = 2 \sum_{i \in S(\nu)} |s_i| \tag{14}$$

is greater than zero and is monotonically increasing as ν increases. Thus, since A^{-1} is positive definite, $-s^T g(\nu)$ is a monotonically decreasing function of ν. Furthermore, since

$$\gamma(\nu) = \frac{-s^T g(\nu)}{s^T A^{-1} s},$$

$\gamma(\nu)$ is also monotonically decreasing. □

An algorithm to determine the optimal point along the descent direction s can now be described. That is, we can determine the global minimizer of the piecewise quadratic function as follows. First compute the vector β, and sort it so that the sequence $\beta_{p(1)}, \ldots, \beta_{p(n)}$ is increasing, where p is the appropriate permutation vector, i.e.,

$$\beta_{p(1)} \leq \beta_{p(2)} \leq \cdots \leq \beta_{p(n)}. \tag{15}$$

Ties can be broken arbitrarily. Examine each successive interval $(\beta_{p(i)}, \beta_{p(i+1)})$ to determine if the minimum of $f_{y,s}$ occurs within it or at the end point $\beta_{p(i+1)}$ as follows. If $\gamma_{p(i)} \leq \beta_{p(i+1)} - \beta_{p(i)}$ then the minimum of $f_{y,s}$ occurs at the minimum of $f_{(p(i),p(i+1))}$, so set $\alpha = \beta_{p(i)} + \gamma_{p(i)}$. Otherwise, if $-s^T g(\beta_{p(i+1)}) \leq 0$ the minimum of $f_{y,s}$ occurs at the breakpoint $\beta_{p(i+1)}$, so set $\alpha = \beta_{p(i+1)}$.

Our line search procedure follows this description *with one important modification*: in order to avoid stopping at a point of nondifferentiability,[3] a *near-optimal* point is computed. Specifically, if the minimum of $f_{y,s}$ occurs at the breakpoint $\beta_{p(i+1)}$, instead of setting $\alpha = \beta_{p(i+1)}$, set $\alpha = \beta_{p(i)} + \tau(\beta_{p(i+1)} - \beta_{p(i)})$, where $\tau = \min(c_2, 1 - \theta/c_1)$, and $0 < c_2 < 1$. This guarantees that $f(y)$ is differentiable at the new point; moreover, the distance to the optimal point along the line goes to zero with θ.

$y = y^0$
$\rho = 1/\|Y\nabla f(y)\|_1$ evaluated at a "typical" value of y
while not optimal **do**
  $\lambda = -(A^{-1}y + A^{-1}b)$
  $\nabla f(y) = -\lambda + \text{sign}(y)$
  $\eta = \rho\|Y\nabla f(y)\|_1 + \sum_i \max((|\lambda_i| - 1), 0)$
  $\theta = c_1\eta/(0.99 + \eta)$
  $R = (\theta I + (1 - \theta)\text{diag}(|\nabla f(y)|))$
  $s = -(|Y|A^{-1} + R)^{-1}(|Y|\nabla f(y))$
  determine α by the line search described above
  $y = y + \alpha s$
enddo

FIG. 3. *The proposed algorithm.*

[3] By (10) the search direction s is not defined if any component of y is zero. Therefore, we avoid such points.

2.3. Implementation details. In this section, we describe an efficient and numerically stable way to implement our algorithm (see Fig. 3). At each iteration, our algorithm requires the computation of a step

$$s = -(|Y|A^{-1} + R)^{-1}(|Y|\nabla f(y))$$

and this computation is the dominant work. However, for reasons of numerical stability, efficiency, and space we do not want to form A^{-1}. (If A is sparse, generally A^{-1} will not be sparse.) Since, by Lemma 2.3, R is nonsingular if y is not optimal, we have the following equivalent linear system of equations:

$$\begin{aligned}
(|Y|A^{-1} + R)s &= -|Y|\nabla f(y), \\
(|Y| + RA)A^{-1}s &= -|Y|\nabla f(y), \\
R^{-\frac{1}{2}}(|Y| + RA)R^{\frac{1}{2}}R^{-\frac{1}{2}}(A^{-1}s) &= -R^{-\frac{1}{2}}(|Y|\nabla f(y)), \\
(|Y| + R^{\frac{1}{2}}AR^{\frac{1}{2}})R^{-\frac{1}{2}}(A^{-1}s) &= -R^{-\frac{1}{2}}(|Y|\nabla f(y)).
\end{aligned}$$

Thus if we solve the symmetric positive definite system

$$(|Y| + R^{\frac{1}{2}}AR^{\frac{1}{2}})v = -R^{-\frac{1}{2}}(|Y|\nabla f(y)), \tag{16}$$

then we can easily compute $s = AR^{1/2}v$. Furthermore, this approach is well suited to sparse problems, since the structure of the matrix in (16) is always the same as that of A, and hence one data structure can be used to store all necessary Cholesky factors. (Note that a similar type of scaling can be used to improve the conditioning of the linear systems to be solved in many other interior-point quadratic programs. See Ye [19] for the general form of these systems.)

When performing the line search, we must compute $(s^k)^T g^k(\beta_j^k)$ at each breakpoint β_j^k that we cross. From (13) and (14), we have

$$-(s^k)^T g^k(\beta_j^k) = -(s^k)^T g^k(0) - \beta_j^k (s^k)^T A^{-1}(s^k) + \sum_{i \in S^k(\beta_j^k)} (2\sigma_i^k s_i^k). \tag{17}$$

(Recall that $S^k(\alpha) = \{i \mid 0 < \beta_i^k \leq \alpha\}$.) Hence we can efficiently obtain $(s^k)^T g^k(\beta_j^k)$ from $(s^k)^T g^k(\beta_{j-1}^k)$ without computing a matrix-vector product. (Note that $(s^k)^T A^{-1} s^k = (s^k)^T R^{1/2} v^k$, where v^k is given by (16).) The work of performing the line search is therefore dominated by the sorting of the breakpoints,[4] which costs $n \cdot \log n$.

3. Global convergence. In this section, we prove that our algorithm converges to the optimal point. We begin by proving some useful bounds.

A notational note: In all subsequent discussion in this section, vector s or s^k refers to the definition given by (10) and (11), unless otherwise noted.

LEMMA 3.1. *There exists $M > 0$ such that for all k, $\|y^k\|_1 \leq M$.*

Proof. Since s is a descent direction, the line search insures that $f(y^k) > f(y^{k+1})$. Thus $\{f(y^k)\}$ is monotonically decreasing. So we have

$$f(y^0) = q_y(y^0) + \|y^0\|_1 \geq f(y^k) = q_y(y^k) + \|y^k\|_1 \geq q_y(-b) + \|y^k\|_1.$$

[4]Even this cost could be reduced, on average, by avoiding the full sort and recursively choosing the minimum breakpoint, i.e., employing a heapsort mechanism.

Thus $\|y^k\|_1 \leq M$, where $M = \|y^0\|_1 + q_y(y^0) - q_y(-b)$. □

COROLLARY 3.2. *$\|\nabla f(y^k)\|_2$, $\|R^k\|_2$, and $\|\lambda^k\|_2$ are bounded above.*

We use this lemma to define the domain of the problem for our nondegeneracy assumption. Let the domain C used in the nondegeneracy assumption be induced by

$$\{y \mid \|y\|_1 \leq M + \epsilon\},$$

where ϵ is an arbitrarily small positive constant. We need ϵ because the proof of superlinear convergence requires nondegeneracy on an open set.

LEMMA 3.3. *$\|s^k\|_2$ is bounded above.*

Proof. We have

$$\begin{aligned}
\|s^k\|_2 &= \|(A^{-1} + |Y^k|^{-1}R^k)^{-1}\nabla f(y^k)\|_2 \\
&\leq \|(A^{-1} + |Y^k|^{-1}R^k)^{-1}\|_2 \|\nabla f(y^k)\|_2 \\
&= \frac{\|\nabla f(y^k)\|_2}{\min\limits_{\|x\|_2=1} x^T(A^{-1} + |Y^k|^{-1}R^k)x} \\
&\leq \frac{\|\nabla f(y^k)\|_2}{\text{min eigenvalue of } (A^{-1})} \\
&= \|\nabla f(y^k)\|_2 \|A\|_2.
\end{aligned}$$

Thus $\|s^k\|_2$ is bounded above. □

Next we show that the function values of the sequence of iterates converge, and the distance between iterates converges to zero.

LEMMA 3.4. *The sequence $\{f(y^k)\}$ is bounded above and below and converges.*

Proof. Since $\{f(y^k)\}$ is monotonically decreasing,

$$f(y^0) \geq f(y^k) = q_y(y^k) + \|y^k\|_1 \geq q_y(y^k) \geq q_y(-b).$$

Thus f is bounded above and below, and hence $\{f(y^k)\}$ converges. □

LEMMA 3.5. *The sequence $\{\|\alpha^k s^k\|_2\} \to 0$.*

Proof. By definition,

$$y^{k+1} = y^k + \alpha^k s^k.$$

Recalling our notation $\sigma^k(\alpha) = \text{sign}(y^k + \alpha s^k)$ and $S^k(\alpha) = \{i \mid 0 < \beta_i^k \leq \alpha\}$, we have

$$\begin{aligned}
f(y^k) - f(y^{k+1}) &= f(y^k) - f(y^k + \alpha^k s^k) \\
&= -(\alpha^k s^k)^T(A^{-1}y^k + A^{-1}b) + \tfrac{1}{2}(\alpha^k s^k)^T A^{-1}(\alpha^k s^k) \\
&\quad + \|y^k\|_1 - \|y^k + \alpha^k s^k\|_1.
\end{aligned}$$

But for all $i \in S^k(\alpha)$,

$$|y_i^k| - |y_i^k + \alpha^k s_i^k| = 2|y_i^k| - \alpha^k s_i^k \sigma_i^k(\alpha),$$

and for $i \notin S^k(\alpha)$,

$$|y_i^k| - |y_i^k + \alpha^k s_i^k| = -\alpha s_i^k \sigma_i^k(\alpha).$$

Therefore, recalling that $\nabla f(y) = A^{-1}y + A^{-1}b + \text{sign}(y)$,

$$f(y^k) - f(y^{k+1}) = -(\alpha^k s^k)^T \nabla f(y^k + \alpha^k s^k) + \frac{1}{2}(\alpha^k s^k)^T A^{-1}(\alpha^k s^k) + 2 \sum_{i \in S^k(\alpha)} |y_i^k|.$$

Since $\sum_{i \in S^k(\alpha)} |y_i^k|$ is nonnegative, and our choice of α^k insures that

$$-(s^k)^T \nabla f(y^k + \alpha^k s^k) \geq 0,$$

we have

$$f(y^k) - f(y^{k+1}) \geq \tfrac{1}{2}(\alpha^k s^k)^T A^{-1}(\alpha^k s^k).$$

Since A^{-1} is positive definite, and $\{f(y^k)\}$ converges, then we must have $\{\|\alpha^k s^k\|_2\} \to 0$. □

Up to this point, none of our results depend on the nondegeneracy assumption; beginning with the next lemma, we will require this assumption. Now we show that under the nondegeneracy assumption, the step s converges to zero. Using this, we can then show that in the limit, complementary slackness is satisfied.

LEMMA 3.6. *Under the nondegeneracy assumption*, $\{\|s^k\|_2\} \to 0$.

Proof. Suppose that $\{\|s^k\|_2\} \not\to 0$. Then Lemma 3.5 implies that a subsequence of $\{\alpha^k\}$ converges to zero. Let $\hat{\alpha}^k = \min(\gamma_0, c_2 \beta^k_{p(1)})$ where p is the permutation vector defined in (15). Note that p depends on the iteration k. Then $\alpha^k \geq \hat{\alpha}^k > 0$ and so zero is a limit point of $\{\hat{\alpha}^k\}$. However, since $g^k(0) = \nabla f(y^k) = -(A^{-1} + |Y^k|^{-1} R^k)(s^k)$, we have

$$\gamma_0 = \frac{-(s^k)^T g^k(0)}{(s^k)^T A^{-1}(s^k)} = \frac{(s^k)^T (A^{-1} + |Y^k|^{-1} R^k)(s^k)}{(s^k)^T A^{-1}(s^k)} \geq 1.$$

Thus zero must be a limit point of $\{\beta^k_{p(1)}\}$. From the definition of s^k, we have $|y_i^k|(\nabla f(y^k)_i + (A^{-1}s^k)_i) = r_i^k s_i^k$, so for each k,

$$\beta^k_{p(1)} = \beta_i^k = \frac{-y_i^k}{s_i^k} = \frac{\sigma_i^k r_i^k}{\nabla f(y^k)_i + (A^{-1}s^k)_i}$$

for some i. Since there are only a finite number of choices of index i, there must be a subsequence of $\{\beta^k_{p(1)}\}$ with $p(1) = j$ for some fixed j. Thus zero must be a limit point of $\{\beta_j^k\}$. Then since $\|s^k\|_2$ is bounded above, a subsequence of $\{y_j^k\}$ must converge to zero. If we assume that the nondegeneracy assumption holds, then the corresponding subsequence of $\{r_j^k\}$ does not have zero as a limit point. Hence a subsequence of it is bounded away from zero. Since $\{\nabla f(y^k)_j\}$ is bounded above, the corresponding subsequence of $(A^{-1}s^k)_j$ must diverge to infinity. However, $\|s^k\|_2$ is bounded above, so this is a contradiction. Therefore we must have $\{\|s^k\|_2\} \to 0$. □

THEOREM 3.7. *Under the nondegeneracy assumption, the sequence* $\{Y^k \nabla f(y^k)\} \to 0$.

Proof. We have

$$\begin{aligned} \| |Y^k| \nabla f(y^k)\|_2 &= \|(|Y^k|\, A^{-1} + R^k)\, s^k\|_2 \\ &\leq \|(|Y^k|\, A^{-1} + R^k)\|_2 \|s^k\|_2 \\ &\leq (\|Y^k\|_2 \, \|A^{-1}\|_2 + \|R^k\|_2)\, \|s^k\|_2. \end{aligned}$$

Since $\|Y^k\|_2$ and $\|R^k\|_2$ are bounded above, if $\|s^k\|_2 \to 0$, we conclude that $\{Y^k \nabla f(y^k)\} \to 0$. □

The next major result is that the sequence $\{y^k\}$ actually converges. Before we can prove this, we need the following two lemmas.

LEMMA 3.8. *Let $\nu \in \mathbb{R}^n$ be such that for all i, $v_i = 1$ or $v_i = -1$. Then the set $Z_\nu = \{y \mid Y(A^{-1}y + A^{-1}b + \nu) = 0\}$ contains a finite number of distinct points.*

Proof. Let $y \in Z_\nu$ and let J be the set of indices of the zero components of y, i.e., $J = \{j \mid y_j = 0\}$. Then since A^{-1} is positive definite, the equation $Y(A^{-1}y + A^{-1}b + \nu) = 0$ uniquely defines the remaining components of y. Hence, Z_ν contains no more points than there are unique subsets of the first n integers, so Z_ν is a finite set. □

The next lemma is standard. See, for example, [15, Note 14.1.2, p. 478], in which Ostrowski [16] is credited.

LEMMA 3.9. *Let $\{y^k\}$ be any bounded sequence of points with the following two properties. The sequence $\{y^k\}$ has a finite number of limit points and $\|y^{k+1} - y^k\|_2 \to 0$. Then the sequence $\{y^k\}$ converges.*

Finally we can show that the sequence of iterates produced by our algorithm converges.

THEOREM 3.10. *Under the nondegeneracy assumption, the sequence $\{y^k\}$ converges.*

Proof. Since Lemma 3.1 implies that the sequence $\{y^k\}$ is bounded, it must have at least one limit point. Let $\hat{y}$ be a limit point of $\{y^k\}$. Thus there is a subsequence of $\{y^k\}$ that converges to $\hat{y}$. Since there are only a finite number of distinct vectors $\text{sign}(y^k)$, there must be an infinite subsequence of this subsequence with $\text{sign}(y^k) = \nu$ for some fixed ν. Hence the corresponding subsequence of $\{Y^k(A^{-1}y^k + A^{-1}b + \nu)\}$ converges to zero, so $\hat{Y}(A^{-1}\hat{y} + A^{-1}b + \nu) = 0$. Since there are only finitely many choices of ν and Lemma 3.8 shows that for each ν the set $Z_\nu = \{y \mid Y(A^{-1}y + A^{-1}b + \nu) = 0\}$ is finite, the sequence $\{y^k\}$ can have only finitely many limit points. Hence Lemmas 3.5 and 3.9 imply that the sequence $\{y^k\}$ converges. □

The next major result is that $\{\lambda^k\}$ converges to a feasible point. We prove this by assuming the contrary and showing that the line search forbids this. First we show that if $|\lambda_j^*| > 1$, then for large enough k, the jth breakpoint will not be crossed during the line search.

LEMMA 3.11. *If $|\lambda_j^*| < 1$, then for large enough k,*

$$\text{sign}(y_j^k) = -\text{sign}(s_j^k) = \text{sign}(\nabla f(y^k)_j).$$

If $|\lambda_j^*| > 1$, then for large enough k,

$$\text{sign}(y_j^k) = -\text{sign}(s_j^k) = \text{sign}(\nabla f(y^k)_j) = -\text{sign}(\lambda_j^k) = -\text{sign}(\lambda_j^*),$$

and during the line search, the jth breakpoint will not be crossed.

Proof. If $|\lambda_j^*| \neq 1$, then since $\nabla f(y^k)_j = -\lambda_j^k + \sigma_j^k$, we have $\{\nabla f(y^k)_j\} \not\to 0$. From the definition of s^k,

$$s_j^k = \frac{-|y_j^k|}{r_j^k}(\nabla f(y^k)_j + (A^{-1}s^k)_j).$$

So since $\{(A^{-1}s^k)_j\} \to 0$, for large enough k, $\text{sign}(s_j^k) = -\text{sign}(\nabla f(y^k)_j)$.

Now suppose $|\lambda_j^*| < 1$. Then for large enough k, $|\lambda_j^k| < 1$, and so $\text{sign}(\nabla f(y^k)_j) = \sigma_j^k$, and hence $\text{sign}(y_j^k) = -\text{sign}(s_j^k) = \text{sign}(\nabla f(y^k)_j)$.

Next suppose $|\lambda_j^*| > 1$. Then for large enough k, we must have $|\lambda_j^k| > 1$. Since $\nabla f(y^k)_j = -\lambda_j^k + \sigma_j^k$ and $|\sigma_j^k| = 1$, we see that for large enough k, $\text{sign}(\nabla f(y^k)_j) = -\text{sign}(\lambda_j^k) = -\text{sign}(\lambda_j^*)$. Hence, $-\text{sign}(s_j^k) = \text{sign}(\nabla f(y^k)_j) = -\text{sign}(\lambda_j^k) = -\text{sign}(\lambda_j^*)$. This says that after some iteration, the sign of s_j^k will remain constant. Since by definition $y^{k+1} = y^k + \alpha^k s^k$, $\{y_j^k\}$ is a monotonic sequence. Since $\{\nabla f(y^k)_j\} \not\to 0$, Theorem 3.7 implies that $\{y_j^k\} \to 0$. In order for this to occur, the sign of s_j^k must be opposite that of y_j^k, otherwise $\{y_j^k\}$ would converge to a nonzero number with the same sign as s_j^k. Thus we must have $\text{sign}(y_j^k) = -\text{sign}(s_j^k) = -\text{sign}(\lambda_j^*)$ and so the jth breakpoint cannot be crossed in the line search. □

The next two lemmas will be used to show that if $|\lambda_i^*| > 1$, and $|\lambda_i^*| > |\lambda_j^*| \neq 1$, then $\beta_i^* < \beta_j^*$. From this we conclude that if $|\lambda_i^*| > 1$ and $\beta_j^k < \beta_i^k$,then for large enough k, $|\lambda_j^k| \geq |\lambda_i^k| > 1$.

LEMMA 3.12. *If* $0 < \lambda_1 < \lambda_2$, *then*

$$\frac{\lambda_1}{1+\lambda_1} < \frac{\lambda_2}{1+\lambda_2}.$$

LEMMA 3.13. *Assume that the nondegeneracy assumption holds. If* $|\lambda_i^*| = 1$ *then any limit points of the sequence* $\{\beta_i^k\}$ *are in the set* $\{-\infty, \infty\}$. *If* $|\lambda_i^*| < 1$ *then any limit points of the sequence* $\{\beta_i^k\}$ *are in the set*

$$\left\{ 1 + \frac{\theta^*|\lambda_i^*|}{1-|\lambda_i^*|},\ 1 - \frac{\theta^*|\lambda_i^*|}{1+|\lambda_i^*|} \right\}.$$

If $|\lambda_i^*| > 1$ *then the limit point of the sequence* $\{\beta_i^k\}$ *is*

$$1 - \frac{\theta^*|\lambda_i^*|}{1+|\lambda_i^*|}.$$

Proof. Suppose $|\lambda_i^*| = 1$, and so using the nondegeneracy assumption, $\{y_i^k\} \not\to 0$. Since

$$\beta_i^k = \frac{-y_i^k}{s_i^k}$$

and $\{s_i^k\} \to 0$, the sequence $\{\beta_i^k\}$ can have only $-\infty$ or $+\infty$ as limit points.

If $|\lambda_i^*| < 1$, then Lemma 3.11 shows that for large enough k, $\text{sign}(\nabla f(y^k)_i) = \sigma_i^k$. Letting $\mu_i^k = \sigma_i^k \text{sign}(\lambda_i^k)$, we can write $\nabla f(y^k)_i = \sigma_i^k(1 - \mu_i^k|\lambda_i^k|)$. We can express β_i^k as

$$\begin{aligned}
\beta_i^k &= \frac{-y_i^k}{s_i^k} = \frac{\sigma_i^k r_i^k}{\nabla f(y^k)_i + (A^{-1}s^k)_i} \\
&= \frac{\sigma_i^k(|\nabla f(y^k)_i| + \theta^k(1 - |\nabla f(y^k)_i|))}{\nabla f(y^k)_i + (A^{-1}s^k)_i} \\
&= \frac{\sigma_i^k(1 - \mu_i^k|\lambda_i^k| + \theta^k\mu_i^k|\lambda_i^k|)}{\sigma_i^k(1 - \mu_i^k|\lambda_i^k| + \sigma_i^k(A^{-1}s^k)_i)} \\
&= 1 + \frac{\theta^k\mu_i^k|\lambda_i^k| - \sigma_i^k(A^{-1}s^k)_i}{1 - \mu_i^k|\lambda_i|^k + \sigma_i^k(A^{-1}s^k)_i}.
\end{aligned}$$

Since $\{(A^{-1}s^k)_i\} \to 0$, $\{\theta^k\} \to \theta^*$, and $\{|\lambda_i^k|\} \to |\lambda_i^*| < 1$, the sequence $\{\beta_i^k\}$ can have as limit points only

$$1 + \frac{\theta^*|\lambda_i^*|}{1 - |\lambda_i^*|} \quad \text{and} \quad 1 - \frac{\theta^*|\lambda_i^*|}{1 + |\lambda_i^*|}.$$

If $|\lambda_i^*| > 1$, then Lemma 3.11 shows that for large enough k, $\operatorname{sign}(\lambda_i^k) = -\sigma_i^k$. Thus $\mu_i^k = -1$. So the limit point of the sequence $\{\beta_i^k\}$ is

$$1 - \frac{\theta^*|\lambda_i^*|}{1 + |\lambda_i^*|}. \qquad \square$$

Now we show that if $|\lambda_j^*| > 1$, then for large enough k, the line search will cause the jth breakpoint to be crossed.

LEMMA 3.14. *Assume that the nondegeneracy assumption holds and that* $|\lambda_j^*| > 1$. *Then for large enough* k,

$$-(s^k)^T g^k(\beta_j^k) > 0. \tag{18}$$

Proof. From (17), we have

$$-(s^k)^T g^k(\beta_j^k) = -(s^k)^T g^k(0) - \beta_j^k (s^k)^T A^{-1}(s^k) + \sum_{i \in S^k(\beta_j^k)} (2\sigma_i^k s_i^k). \tag{19}$$

Using the fact that $-(s^k)^T g^k(0) = -(s^k)^T \nabla f(y^k) = (s^k)^T (A^{-1} + |Y^k|^{-1} R^k)(s^k)$, we can rewrite the right-hand side of (19) as

$$(1 - \beta_j^k)(s^k)^T A^{-1}(s^k) + (s^k)^T (|Y^k|^{-1} R^k)(s^k) + \sum_{i \in S^k(\beta_j^k)} (2\sigma_i^k s_i^k). \tag{20}$$

For large enough k, Lemma 3.13 shows that $\beta_j^k < 1$, so the first term in (20) is greater than zero. We can express the second and third terms as

$$\sum_{i \in S^k(\beta_j^k)} \left(\frac{r_j^k |s_j^k|}{|y_j^k|} - 2 \right) |s_j^k| + \sum_{i \notin S^k(\beta_j^k)} \frac{r_i^k}{|y_i^k|} (s_i^k)^2. \tag{21}$$

The second sum in (21) is obviously greater than zero. Thus the only thing remaining to show is that the first sum in (21) is greater than zero. We can simplify the summand in the first sum as follows:

$$\left(\frac{r_i^k |s_i^k|}{|y_i^k|} - 2 \right) |s_i^k| = (|\nabla f(y^k)_i + (A^{-1}s^k)_i| - 2)|s_i^k|. \tag{22}$$

For large enough k, Lemmas 3.12 and 3.13 show that if $i \in S^k(\beta_j^k)$ then $|\lambda_i^k| \geq |\lambda_j^k| > 1$. Hence Lemma 3.11 implies that $|\nabla f(y^k)_i| = (1 + |\lambda_i^k|) > 2$. Since $\{(A^{-1}s^k)\} \to 0$, for large enough k,

$$(|\nabla f(y^k)_i + (A^{-1}s^k)_i| - 2) > 0.$$

Therefore, for large enough k, each term in the first sum in (21) is greater than zero, and the proof is complete. $\square$

Now we can prove that λ is feasible and derive some corollaries that we will use to prove superlinear convergence and to show that the step length converges to unity.

THEOREM 3.15. *Under the nondegeneracy assumption,* $|\lambda_i^*| \leq 1$ *for all* i.

Proof. Suppose $|\lambda_i^*| > 1$ for some i. Lemma 3.11 shows that for large enough k, the ith breakpoint cannot be crossed in the line search. Lemma 3.14 shows that for large enough k,

$$-(s^k)^T g^k(\beta_i^k) > 0,$$

and so our line search would cause the ith breakpoint to be crossed. These are contradictory statements and hence for all i, $|\lambda_i^*| \leq 1$. □

COROLLARY 3.16. *Under the nondegeneracy assumption,* $\{\theta^k\} \to 0$.

COROLLARY 3.17. *Under the nondegeneracy assumption, if* $y_i^* = 0$ *then* $\beta_i^* = 1$, *and if* $y_i^* \neq 0$ *then* $\beta_i^* = \pm\infty$.

Proof. The first statement follows immediately from Lemma 3.13, the definition of θ, and Theorems 3.7 and 3.15. The second statement follows from Theorem 3.7 and Lemma 3.13. □

4. Superlinear convergence. In this section we establish that under the nondegeneracy assumption, the sequence $\{y^k\}$ produced by our algorithm converges to y^* superlinearly. Consider the following finite set $\mathcal{F}$ of functions

$$F_\nu(y) = Y(A^{-1}y + A^{-1}b + \nu),$$

where $\nu \in \mathbb{R}^n$ is defined as

$$\nu_i = \begin{cases} +1 \text{ or } -1 & \text{if } y_i^* = 0, \\ \text{sign}(y_i^*) & \text{otherwise.} \end{cases}$$

Each function F_ν is twice continuously differentiable and, furthermore, $F_\nu(y^*) = 0$.

The Jacobian of $F_\nu(y)$ is $J_\nu(y) = YA^{-1} + G_\nu(y)$, where $G_\nu(y) = \text{diag}(A^{-1}y + A^{-1}b + \nu)$. Note that the nondegeneracy assumption implies that $J_\nu(y)$ is nonsingular. The Newton step at y^k for finding a zero of F_ν is

$$(Y^k A^{-1} + G_\nu(y^k))s_N^k = -F_\nu(y^k).$$

Lemma 3.11 shows that for large enough k, $F_{\sigma^k} \in \mathcal{F}$, and hence our search direction s^k satisfies

$$(Y^k A^{-1} + \Sigma^k R^k)s^k = -F_{\sigma^k}(y^k),$$

where $\Sigma^k = \text{diag}(\sigma^k)$. Thus s^k is very similar to a Newton step at y^k and, in fact, we will show that s^k converges to a Newton step. But first we state a more general result about superlinear convergence of a family of functions. This result follows easily from Theorem 3.4 in Dennis and Moré [6].

THEOREM 4.1. *Let* $\mathcal{F} = \{F_\nu : \mathbb{R}^n \to \mathbb{R}^n\}$ *be a finite set of functions satisfying the following assumptions:*

- *Each* F_ν *is continuously differentiable in an open convex set* C.
- *There is a* y^* *in* C *such that* $F_\nu(y^*) = 0$ *and* $\nabla F_\nu(y^*)$ *is nonsingular.*
- *There is a constant* κ *such that for all* $F_\nu \in \mathcal{F}$,

$$\|\nabla F_\nu(y) - \nabla F_\nu(y^*)\| \leq \kappa\|y - y^*\|$$

for $y \in C$.

Let $\{W^k\}$ in $L(\mathbb{R}^n)$ be a sequence of nonsingular matrices. Suppose that for some y^0 in C the sequence

$$y^{k+1} = y^k - (W^k)^{-1} F_{\nu^k}(y^k), \qquad k = 0, 1, \ldots,$$

remains in C and converges to y^*, and that $y^k \neq y^*$ for $k > 0$. Then, if

$$\{\|W^k - \nabla F_{\nu^k}(y^*)\|\} \to 0, \tag{23}$$

$\{y^k\}$ converges superlinearly to y^*.

Now we show that our set of functions and the sequence generated by our algorithm satisfy the hypotheses of Theorem 4.1. For the convex open set, we take the region

$$C = \{y \mid \|y\|_1 < M + \epsilon\},$$

where M is as in Lemma 3.1 and ϵ is an arbitrarily small positive constant. We have seen that the first two assumptions hold. The next lemma shows that the third one holds.

LEMMA 4.2. *There is a constant κ such that for all $F_\nu \in \mathcal{F}$,*

$$\|\nabla F_\nu(y) - \nabla F_\nu(y^*)\|_1 \leq \kappa \|y - y^*\|_1$$

for $y \in C$.

Proof. Set $\delta = \max(M + \epsilon, \|b\|_1)$. We have

$$\begin{aligned}
\|\nabla F_\nu(y) - \nabla F_\nu(y^*)\|_1 &= \|Y(A^{-1}y + A^{-1}b + \nu) - Y^*(A^{-1}y^* + A^{-1}b + \nu)\|_1 \\
&\leq \|Y - Y^*\|_1 \, \|A^{-1}y\|_1 \; + \; \|Y^* A^{-1}\|_1 \, \|y - y^*\|_1 \\
&\qquad + \; \|Y - Y^*\|_1 \, \|A^{-1}b + \nu\|_1 \\
&\leq (3 \, \|A^{-1}\|_1 \, \delta + \|\nu\|_1) \, \|y - y^*\|_1 \\
&= \kappa \, \|y - y^*\|_1,
\end{aligned}$$

where $\kappa = 3 \, \|A^{-1}\|_1 \, \delta + n$. Thus we have the desired result. □

Before we can prove that (23) holds, we must show that the step length converges to one. The next lemma shows that for any fixed $\alpha > 1$, for large enough k, a step of length α takes us beyond the minimum of $f_{y,s}$.

LEMMA 4.3. *Assume that the nondegeneracy assumption holds and that $\alpha > 1$. Then for large enough k,*

$$\gamma^k(\alpha) = \frac{-(s^k)^T g(\alpha, y^k, s^k)}{(s^k)^T A^{-1}(s^k)} < 0.$$

Proof. From (13) and (14), we have

$$\gamma^k(\alpha) = \frac{1}{(s^k)^T A^{-1}(s^k)} \left(-(s^k)^T g^k(0) - \alpha (s^k)^T A^{-1}(s^k) - \sum_{i \in S^k(\alpha)} (-2\sigma_i^k s_i^k) \right). \tag{24}$$

Using the fact that $-(s^k)^T g^k(0) = -(s^k)^T \nabla f(y^k) = (s^k)^T (A^{-1} + |Y^k|^{-1} R^k)(s^k)$, we can rewrite (24) as

$$\gamma^k(\alpha) = (1 - \alpha) + \frac{(s^k)^T (|Y^k|^{-1} R^k)(s^k)}{(s^k)^T A^{-1}(s^k)} + \sum_{i \in S^k(\alpha)} \frac{-2\sigma_i^k s_i^k}{(s^k)^T A^{-1}(s^k)}.$$

Reorganizing, we get

$$\gamma^k(\alpha) = (1-\alpha) + \sum_{i\in S^k(\alpha)} \left(\frac{r_i^k|s_i^k|}{|y_i^k|} - 2\right) \frac{|s_i^k|}{(s^k)^T A^{-1}(s^k)} + \sum_{i\notin S^k(\alpha)} \frac{r_i^k}{|y_i^k|} \frac{(s_i^k)^2}{(s^k)^T A^{-1}(s^k)}$$

$$= (1-\alpha) + \sum_{i\in S^k(\alpha)} (|\nabla f(y^k)_i + (A^{-1}s^k)_i| - 2) \frac{|s_i^k|}{(s^k)^T A^{-1}(s^k)}$$

$$+ \sum_{i\notin S^k(\alpha)} \frac{r_i^k}{|y_i^k|} \frac{(s_i^k)^2}{(s^k)^T A^{-1}(s^k)}.$$

If $i \in S^k(\alpha)$, Theorem 3.15 and our nondegeneracy assumption show that for large enough k, $|\lambda_i^k| < 1$, and hence $|\nabla f(y^k)_i| = |(\sigma_i^k - \lambda_i^k)| < 2$. Furthermore, since $\{(A^{-1}s^k)\} \to 0$, for large enough k,

$$(|\nabla f(y^k)_i + (A^{-1}s^k)_i| - 2) < 0.$$

Thus, we can bound $\gamma^k(\alpha)$ as follows:

$$\gamma^k(\alpha) \le (1-\alpha) + \sum_{i\notin S^k(\alpha)} \frac{r_i^k}{|y_i^k|} \frac{(s_i^k)^2}{(s^k)^T A^{-1}(s^k)}$$

$$\le (1-\alpha) + \sum_{i\notin S^k(\alpha)} \left(\frac{r_i^k}{|y_i^k|}\right) \left(\frac{1}{\text{min eigenvalue of } A^{-1}}\right) \left(\frac{(s_i^k)^2}{\|s^k\|_2^2}\right)$$

$$\le (1-\alpha) + \|A\|_2 \sum_{i\notin S^k(\alpha)} \frac{r_i^k}{|y_i^k|}.$$

For large enough k, if $i \notin S^k(\alpha)$, then $y_i^k \not\to 0$ and hence $r_i^k \to 0$. Thus $\sum_{i\notin S^k(\alpha)} (r_i^k/|y_i^k|)$ converges to zero as $k \to \infty$. So, since $\alpha > 1$, for large enough k, $\gamma^k(\alpha) < 0$. □

THEOREM 4.4. *Under the nondegeneracy assumption*, $\{\alpha^k\} \to 1$.

Proof. Corollary 3.17 implies that $\{\beta_{p(1)}^k\} \to 1$ or $\{\beta_{p(1)}^k\} \to \infty$ and Corollary 3.16 implies that $\{\theta^k\} \to 0$. By definition, $\alpha^k \ge (1 - \theta^k/c_2)\beta_{p(1)}^k$, hence $\{\alpha\}$ cannot have a limit point that is less than 1. Furthermore, the properties of the line search, combined with Lemma 4.3, show that for any ϵ there exists $k(\epsilon)$, such that for $k > k(\epsilon)$, α^k cannot be greater than $1 + \epsilon$. Thus $\{\alpha^k\} \to 1$. □

The last thing necessary to prove superlinear convergence is to show that (23) holds and we show this in the next lemma.

LEMMA 4.5. *Let*

$$W^k = \frac{1}{\alpha^k}(Y^k A^{-1} + \Sigma^k R^k).$$

Then, under the nondegeneracy assumption, $\|W^k - \nabla F_{\sigma^k}(y^*)\|_2 \to 0$.

Proof. Notice that $G_{\sigma^k}(y^k) = \text{diag}(\nabla f(y^k))$. From the definitions of W^k, $\nabla F_{\sigma^k}(y^*)$, and R^k, we have

$$\|W^k - \nabla F_{\sigma^k}(y^*)\|_2 = \left\| \frac{1}{\alpha^k}(Y^k A^{-1} + \Sigma^k R^k) - (Y^* A^{-1} + G_{\sigma^k}(y^*)) \right\|_2$$

$$\leq \left\| \left(\left(\frac{1}{\alpha^k} \right) Y^k - Y^* \right) A^{-1} \right\|_2 + \left\| \frac{1}{\alpha^k} (\Sigma^k \Gamma^k G_{\sigma^k}(y^k)) - G_{\sigma^k}(y^*) \right\|_2 + \frac{\theta^k}{\alpha^k} \| \Sigma^k (I - |G_{\sigma^k}(y^k)|) \|_2,$$

where $\Gamma^k = \text{diag}(\text{sign}(\nabla f(y^k)))$. Theorem 4.4 shows that $\{\alpha^k\} \to 1$, and Corollary 3.16 shows that $\{\theta^k\} \to 0$. Furthermore, since $\{y^k\} \to \{y^*\}$ and $\|\nabla f(y^k)\|_2$ is bounded above, the first and third terms on the right-hand side of the above inequality converge to zero. Hence it suffices to show that

$$\left\| \frac{1}{\alpha^k} (\Sigma^k \Gamma^k G_{\sigma^k}(y^k)) - G_{\sigma^k}(y^*) \right\|_2 \to 0 \tag{25}$$

to obtain the desired result. But (25) follows immediately from Lemmas 3.7 and 3.11. □

Thus the conditions of Theorem 4.1 hold and we have the following theorem.

THEOREM 4.6. *Under the nondegeneracy assumption, the sequence* $\{y^k\}$ *generated by our algorithm converges superlinearly to* y^*.

Our numerical experiments suggest that our algorithm may indeed be quadratically convergent in the nondegenerate case; however, we have not been able to establish this yet. It is easy to see what needs to be proved. Under the assumptions of Theorem 4.1, if

$$\|W^k - \nabla F_{\nu^k}(y^*)\| = O(\|y^k - y^*\|), \tag{26}$$

then $\{y^k\}$ converges quadratically to y^*. It is straightforward to show that $\|Y^k - Y^*\| = O(\|y^k - y^*\|)$, $\|\Sigma^k \Gamma^k G_{\sigma^k}(y^k) - G_{\sigma^k}(y^*)\| = O(\|y^k - y^*\|)$, and $\theta^k = O(\|y^k - y^*\|)$. Hence from the proof of Lemma 4.5 it suffices to show that $1 - \alpha^k = O(\|y^k - y^*\|)$. The rate at which $\alpha^k \to 1$ depends on the rates at which γ^k decreases and $\beta^k \to 1$.

5. Numerical results.

5.1. The test problems. We generate test problems of the form (1) in the manner suggested by Moré and Toraldo [12]. They describe how to generate problems, varying four parameters: n, the number of variables; *lcnd*, the logarithm base 10 of the condition number of A; *nb*, the number of variables at their bound at the solution x^*; and *ymag*, the magnitude of the nonzero components of y^*.

To generate a test problem whose solution has certain properties, choose A to have the desired properties, generate x^* and $y^* = \nabla q_x(x^*)$ such that either x_i^* is at a bound (i.e., $|x_i^*| = 1$) or $y_i^* = 0$, but not both, and then set $b = -Ax^* + y^*$. In particular, set

$$A = QDQ \quad \text{where } Q = I - \frac{2}{\|y\|^2} yy^T,$$

D is a diagonal matrix with

$$d_{ii} = 10^{k_i \cdot lcnd}, \quad k_i = \frac{i-1}{n-1}, \quad i = 1, \ldots, n,$$

and the components of y are randomly generated in the interval $(-1, 1)$. Thus A is a positive definite matrix with condition number 10^{lcnd}.

Given *nb*, the number of variables at bounds at the solution, generate x^* as follows. Let B be the index set identifying the components of y that are zero at the solution:

$i \in B \iff y_i^* = 0$. Let B^c be the complementary set. First choose B and B^c by generating a random number μ_i in $(0,1)$ for each $i = 1,\ldots,n$ and include i in B^c if $\mu_i < nb/n$. Then choose x^* by setting those components in B^c randomly to $+1$ or -1, and selecting the remaining components by randomly generating x_i in $(-1,1)$.

Generate y^* as follows. If $i \in B$, set $y_i^* = 0$. Otherwise randomly generate μ_i in $(-1,1)$ and ν_i in $(0,1)$ and set

$$y^* = \text{sign}(\mu_i) \times 10^{-\nu_i \cdot ymag}. \tag{27}$$

Then, by setting $b = y^* - Ax^*$, we have a problem with the desired characteristics.

5.2. Numerical results. In this section, we examine the numerical behavior of our algorithm. Our implementation is in Pro-Matlab[5] and all experiments were performed using a collection of Sun Sparcstations.

Our implemention follows the algorithm described in §2. We use a single stopping criterion based on the change in objective function value: the algorithm is terminated at y^{k+1} if

$$|f(y^{k+1}) - f(y^k)| \leq \text{tol} \cdot (1 + |f(y^k)|). \tag{28}$$

We set tol $= 10^{-15}$ for all the experiments except for the "low-precision" results where we use tol $= 10^{-8}$.

As our starting point, we choose the origin, i.e., $x^0 = 0$. Empirically, we determine that $c_1 = 10^{-3}$ and $c_2 = 0.90$ are reasonable choices of these parameters and we use them in our tests.

To capture the behavior of the algorithm, we vary each of the problem parameters, in turn, while keeping the others fixed. For the results quoted in the first six tables, we fix $n = 100$, restrict *lcnd* to the values 0, 3, 6, 9, and 12, and assign to nb the values 10, 50, and 90. We restrict $y$mag to be 1, 3, 6, 9, or 12, where the magnitude of the nonzero components of y is about $10^{-y\text{mag}}$. Therefore, the test problems become increasingly near-degenerate as $y$mag increases.

First, in order to compare the results of [8] we consider problems run to low-accuracy; i.e., tol $= 10^{-8}$. We consider 10 problems for each set of problem parameters; therefore, Tables 1–3 represent a total of 750 test problems. We report the average, maximum, and minimum number of iterations required to achieve the convergence criterion in (28).

The iteration averages in Tables 1–3 can be compared to the results given in [8] in which problems with identical characteristics (though not identical problems) were generated to test the feasible-point algorithm proposed in [8]. The stopping criteria used in both cases are comparable, as is the cost of an iteration, since in both algorithms the cost of an iteration is dominated by the solution of linear systems with identical structures. Inspection reveals that our proposed algorithm requires fewer average iterations in 71 out of 75 cases with an average differential of about 3; therefore, we feel confident in concluding that our method is at least competitive with [8] in the low-accuracy setting. (This comes with the caveat that while the method proposed in [8] maintains feasibility, our new method is only near-feasible on termination. However, the simple strategy of setting all infeasible variables to their nearest bounds upon termination is possible: on our test set this technique did not significantly increase the function value in any case.)

[5]Our results involving sparse matrices were obtained using an experimental version of Matlab [7] in which sparse matrices can be easily manipulated.

TABLE 1
Iterations for nb = 10 (*low accuracy,* tol $= 10^{-8}$).

	ymag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	10	9	11	11.5	11	12	11.5	11	12	12	11	13	11.9	11	13
3	10.6	9	12	10.7	10	12	11.7	11	13	11.6	11	12	11.9	10	13
6	9.9	9	11	11.2	10	13	11.8	11	13	11.4	10	12	11.8	11	13
9	9.7	9	10	10.3	9	12	10.9	10	12	11.2	11	12	11.6	10	12
12	9.3	8	11	9.9	9	13	10.6	9	14	11.6	9	12	11	10	12

TABLE 2
Iterations for nb = 50 (*low accuracy,* tol $= 10^{-8}$).

	ymag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	9.6	9	11	10.7	10	11	10.8	10	12	10.8	10	12	10.8	10	12
3	9.4	8	10	10.6	10	13	10.8	10	12	11.4	11	12	11.5	10	15
6	8.8	8	9	10.1	9	11	10.9	10	12	11.1	10	12	11.5	10	16
9	8.3	7	9	9.7	9	11	11.4	10	15	11.1	10	13	10.4	10	11
12	8.1	7	9	9.4	9	10	10.2	9	11	10.2	9	11	10.1	9	12

TABLE 3
Iterations for nb = 90 (*low accuracy,* tol $= 10^{-8}$).

	ymag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	8.3	7	9	9.2	8	10	8.8	8	10	8.8	8	9	8.4	8	9
3	8.0	7	9	9.8	9	11	10.5	8	16	9.7	8	11	9.5	9	10
6	8.1	7	9	12.3	8	23	10.2	9	13	9.7	9	11	9.7	9	11
9	7.2	6	8	10.2	8	15	10.6	9	16	9.3	8	10	9.8	8	15
12	7	6	8	9.9	8	18	9.8	8	14	9.7	9	10	9.2	8	11

Due to the second-order nature of our algorithm it is usually possible to obtain significantly greater accuracy at reasonable cost. Our next experiments, reported in Tables 4–6, involve exactly the same test problems as above, except that now tol $= 10^{-15}$; again, condition (28) is our sole stopping criterion.

In most cases the step from low accuracy (tol $= 10^{-8}$) to high accuracy (tol $= 10^{-15}$) involves only a modest increase in effort. The better-conditioned problems require one to two extra iterations. As *lcnd* and *ymag* increase, the number of extra iterations increases to about four or five, typically. The maximum number of iterations required by any problem, out of 750, is 41; the worst average is 20.4. Most of the problems require 17 or fewer iterations.

In our test set the accuracy achieved in the objective function value, $q_x(\bar{x})$, where $\bar{x}$ indicates the computed solution, is always acceptable. Specifically, out of 750 test problems the following bound is achieved:

$$\max \left| \frac{q_x(\bar{x}) - \text{opt}}{\text{opt}} \right| \leq 10^{-10}, \tag{29}$$

TABLE 4
Iterations for nb = 10 (*high accuracy,* tol $= 10^{-15}$).

	ymag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	11.4	11	12	13.5	13	15	15.7	14	18	17.2	16	18	16.1	15	17
3	12.6	11	15	12.5	12	14	15.2	13	18	16	14	18	16.4	14	18
6	12.9	11	15	13.7	12	16	15.5	13	19	16.6	15	21	16.8	14	18
9	13.5	13	15	13.6	11	16	14.3	11	16	15.8	14	20	16.5	14	17
12	12.4	11	15	13.2	11	15	13.7	11	17	17.3	14	28	15.4	12	18

TABLE 5
Iterations for nb = 50 (*high accuracy,* tol $= 10^{-15}$).

	ymag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	10.8	10	12	13.7	13	15	15.9	15	17	16.3	15	17	16.1	15	17
3	11.1	10	12	13	12	15	15.6	14	17	16.5	15	18	16.7	15	19
6	11.9	10	14	12.5	11	15	15.2	14	16	16	15	18	16.5	14	22
9	12.5	10	15	13.1	12	15	15.7	14	19	17.0	14	21	15.8	14	17
12	12.3	11	13	12.1	11	14	14.3	13	17	15.6	14	17	16.2	13	20

TABLE 6
Iterations for nb = 90 (*high accuracy,* tol $= 10^{-15}$).

	ymag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	10.1	9	13	12.2	11	14	14.1	13	15	14	13	15	13.9	13	15
3	9.2	9	10	12.9	11	17	20.4	14	41	16.6	14	22	15.9	15	22
6	10.4	9	12	14.8	11	25	17.5	13	31	16.8	15	21	16.2	14	21
9	10.3	9	11	13.6	11	19	15.6	13	22	16.7	14	22	16	13	21
12	11.2	10	13	13	10	20	15.6	13	23	19.3	15	32	16.3	12	24

where opt is the true optimal value, opt $\neq 0$. Moreover, in the vast majority of cases we achieve

$$\max \left| \frac{q_x(\bar{x}) - \text{opt}}{\text{opt}} \right| \leq 10^{-15}, \tag{30}$$

which is essentially full accuracy in the objective function value. Of course, the accuracy achieved in x varies depending on the conditioning of the problem. The worst feasibility result, over all 750 test cases, is

$$\max\{\max(|\bar{x}_i| - 1, 0)\} = 10^{-5}.$$

In all cases, setting infeasible variables to their nearest bound upon termination changed the objective function value only mildly; our worst-case bound (29) is maintained after this correction as well as the observation that (30) holds in the vast majority of the cases.

In order to test the sensitivity of our algorithm to problem size, we consider larger test cases, involving sparse matrices, and present our results in Tables 7–11. Thanks to Cleve Moler of The Mathworks, Inc., we were able to perform our experiments using

an experimental version of Matlab, in which sparse matrices are easily generated and manipulated [7].

In our sparse experiments we strive for high accuracy, i.e., tol $= 10^{-15}$, and we hold the percentage of bound constraints, *nb*, fixed at 50 %. The matrices are generated using a Matlab subroutine SPRAND supplied to us by Rob Schreiber. Given the density of the matrix (dens) as well as *lcnd* and the base-10 exponent of the condition number of the matrix, SPRAND produces a sparse symmetric positive definite matrix with the given condition number and a random sparsity pattern with number of nonzeros approximately equal to dens $\times n^2$. In our tests dens $= \frac{5}{n}$ and *lcnd* $= 4, 8$. Our test suite consists of 5 test problems for each setting of the problem parameters, yielding a total of 100 test problems.

TABLE 7
Sparse problems, iterations for $n = 100$.

	*y*mag					
	1			5		
lcnd	avg	min	max	avg	min	max
4	11.6	10	13	15.4	13	18
8	12.4	12	13	15	13	20

TABLE 8
Sparse problems, iterations for $n = 200$.

	*y*mag					
	1			5		
lcnd	avg	min	max	avg	min	max
4	12.4	11	14	21.4	17	31
8	12.4	12	13	21.2	14	30

TABLE 9
Sparse problems, iterations for $n = 500$.

	*y*mag					
	1			5		
lcnd	avg	min	max	avg	min	max
4	14.4	13	17	21.4	16	29
8	14	13	16	29.6	18	45

TABLE 10
Sparse problems, iterations for $n = 1000$.

	*y*mag					
	1			5		
lcnd	avg	min	max	avg	min	max
4	14.8	14	16	21.8	19	24
8	15.2	17	13	25.4	22	31

TABLE 11
Sparse problems, iterations for $n = 2000$.

	ymag					
	1			5		
lcnd	avg	min	max	avg	min	max
4	15.2	14	16	28.6	23	33
8	16.2	14	18	33.4	23	47

The average number of iterations grows rather mildly with n. For example, in the moderately ill conditioned setting $lcnd = 4$, $ymag = 5$, the average number of iterations goes from $15.4(n = 100)$ to $21.4(n = 500)$ to $28.6(n = 2000)$.

The accuracy achieved on this set of large sparse problems is quite good. In particular, essentially full accuracy in the objective function value is achieved in every case:

$$\max \left| \frac{q_x(\bar{x}) - \text{opt}}{\text{opt}} \right| \leq 10^{-15}, \tag{31}$$

where $\bar{x}$ is the computed solution and opt is the true optimal value; feasibility was also respectable:

$$\max \{\max (|\bar{x}_i| - 1, 0)\} = 10^{-9}.$$

6. Conclusions. This paper presents a new algorithm for solving box-constrained convex quadratic programs. The method shows promise: beyond global and superlinear convergence results, the numerical experiments indicate practical potential. Specifically, high accuracy can usually be achieved with a modest number of iterations.

The real promise of this approach is in the large-scale setting where questions of exploiting sparsity or parallelism can be centered on the Cholesky factorization alone. Work outside of the factorization/solve is bounded by $\text{nnz}(A) + n \cdot \log n$, where $\text{nnz}(A)$ is the number of nonzeros of A. This work is usually negligible compared to the factorization.The Cholesky factorization is a standard linear algebra task in both the sparse and parallel settings; therefore, we need only "plug into" a standard routine to achieve efficiency.

Further research needs to be done. For example, we believe the degeneracy assumption can be greatly relaxed without weakening the theoretical properties; the question of quadratic convergence should be resolved (probably in the affirmative); more work is needed on the handling of different bounds, including one-sided bounds.

Despite the promising results of this paper with respect to our new algorithm, the most important contribution may lie elsewhere. Specifically, the *ideas* underpinning this algorithm are new (or are used in a novel way) and their full domain of applicability is unknown. To summarize, the basic underlying ideas are: the transformation of a constrained problem to a piecewise differentiable problem,[6] the notion of a Newton process for this nondifferentiable function, the definition of a descent direction in combination with an efficient line search procedure. We expect that many more problems can be approached in this way. For example, the successful l_1 algorithm in [7] also follows these lines.

[6]The lack of penalty parameter (or, equivalently, penalty parameter equal to unity) is due to two things. First, it is not hard to see that minimization of a quadratic function subject to finite box-constraints on every variable is equivalent to minimization of an unconstrained piecewise quadratic function with an easily computed penalty parameter. Second, the homogeneous unit bounds in (1) yield a unit penalty parameter.

Acknowledgment. The authors thank Mike Todd for pointing out a flaw in the proof of convergence in a previous draft and Yuying Li for numerous helpful discussions.

REFERENCES

[1] Å. BJÖRCK, *A direct method for sparse least squares problems with lower and upper bounds*, Numer. Math., 54 (1988), pp. 19–32.

[2] T. F. COLEMAN AND A. R. CONN, *Second-order conditions for an exact penalty function*, Math. Programming, 19 (1980), pp. 178–185.

[3] T. F. COLEMAN AND L. A. HULBERT, *A direct active set algorithm for large sparse quadratic programs with simple bounds*, Math. Programming, 45 (1989), pp. 373–406.

[4] T. F. COLEMAN AND Y. LI, *A globally and quadratically convergent affine scaling method for linear l_1 problems*, Math. Programming, 56, Ser. A (1992), pp. 189–222.

[5] R. S. DEMBO AND U. TULOWITZKI, *On the minimization of quadratic functions subject to box constraints*, Tech. Rep., B 71, School of Organization and Management, Yale University, New Haven, CT, 1983.

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

[7] J. GILBERT, C. MOLER, AND R. SCHREIBER, *Sparse matrices in matlab: Design and implementation*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 333–356.

[8] C.-G. HAN, P. M. PARDALOS, AND Y. YE, *Computational aspects of an interior point algorithm for quadratic programming problems with box constraints*, Large-Scale Numerical Optimization, T. F. Coleman and Y. Li, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990, pp. 92–112.

[9] J. J. JÚDICE AND F. M. PIRES, *Direct methods for convex quadratic programs subject to box constraints*, Departamento de Matemática, Univ. de Coimbra, Coimbra, Portugal, 1989.

[10] N. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[11] P. LOTSTEDT, *Solving the minimal least squares problem subject to bounds on the variables*, BIT, 24 (1984), pp. 206–224.

[12] J. J. MORÉ AND G. TORALDO, *Algorithms for bound constrained quadratic programming problems*, Numer. Math., 55 (1989), pp. 377–400.

[13] D. P. O'LEARY, *A generalized conjugate gradient algorithm for solving a class of quadratic programming problems*, Linear Algebra Appl., 34 (1980), pp. 371–399.

[14] U. ÖREBORN, *A Direct Method for Sparse Nonnegative Least Squares Problems*, Ph.D. thesis, Dept. of Mathematics, Linköping Univ., Linköping, Sweden, 1986.

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

[16] A. OSTROWSKI, *Solution of Equations and Systems of Equations*, Academic Press, New York, 1966.

[17] E. K. YANG AND J. W. TOLLE, *A class of methods for solving large convex quadratic programs subject to box constraints*, Tech. Rep., Dept. of Operations Research, Univ. of North Carolina, Chapel Hill, NC, 1988.

[18] Y. YE, *An extension of Karmarkar's algorithm and the trust region method for quadratic programming*, in *Progress in Mathematical Programming*, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 49–63.

[19] ———, *Interior point algorithms for quadratic programming*, Tech. Rep. 89-29, Dept. of Management Sciences, Univ. of Iowa, Iowa City, IA, 1989.

[20] Y. YE AND E. TSE, *An extension of Karmarkar's projective algorithm for convex quadratic programming*, Math. Programming, 44 (1989), pp. 157–179.