A Globally and Superlinearly Convergent Algorithm for Convex Quadratic Programs with Simple Bounds*

Thomas F. Coleman
Laurie A. Hulbert

TR 90-1092
February 1990

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*Research 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 DE-FG02-86ER25013.A000 and by the Computational Mathematics Program of the National Science Foundation under grant DMS-8706133.

A Globally and Superlinearly Convergent Algorithm for Convex Quadratic Programs with Simple Bounds [†]

Thomas F. Coleman
Computer Science Department
Cornell University
Ithaca, New York 14853

Laurie A. Hulbert
Center For Applied Mathematics
Cornell University
Ithaca, New York 14853

March 1990

Abstract: We present a globally and superlinearly convergent algorithm for solving convex quadratic programs with simple bounds. 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. 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 our algorithm is suitable for solving large sparse problems and for implementation on parallel computers. We implemented our algorithm and tested it on a sequential computer on a variety of dense problems, and we present numerical results which show that our algorithm solves many problems quickly.

Keywords: quadratic programming, interior point methods, simple bounds, box constraints, large sparse minimization

[†] Research 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.

1 Introduction

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

$$\begin{array}{c} \min \frac{1}{2}x^T Ax + b^T x \\ l \leq x \leq u, \end{array} \tag{1}$$

where A is an $n \times n$ positive definite symmetric matrix. Previous algorithms for solving this problem basically fall into two categories: active set algorithms and interior point algorithms. Active set algorithms solve a sequence of equality constrained quadratic programs. They are conceptually simple but their implementation requires complicated data structure manipulations, especially for large sparse problems. On simply bound programs, the major work at each step is the Cholesky factorization of a matrix that is smaller than the original matrix A. Hence these algorithms can solve many quadratic programs in time proportional to three or four Cholesky factorizations of A, although their performance varies quite a bit depending on the characteristics of the particular problems [3,12]. Recent active set algorithms for this and similar problems have been proposed by 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 [15].

Interior point algorithms for quadratic programs are typically based on affine scaling, path following, potential reduction, or projection techniques and are in general simpler to implement than active set methods because they require less data structure manipulation. The major work at each step is a Cholesky factorization. Their simplicity makes these methods attractive for parallel computers. Interior point algorithms for quadratic programs, and in particular, polynomial algorithms, have received much interest recently with the advent of Karmarkar's interior point algorithm for linear programming [10]. Many of the new approaches for linear programs have been extended to quadratic programs. For a discussion of recent interior point algorithms for this and other quadratic programs, see the survey paper by Ye [17]. Most of these interior point algorithms have polynomial time bounds, but their asymptotic rates of convergence have not been studied. The affine scaling method proposed by Ye [16], which is similar to but simpler than the polynomial algorithm of Ye and Tse [18] 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; however, their algorithm generally requires more than 10 steps and hence, on a sequential computer, would frequently be slower than an active set method.

We present an algorithm that is neither an active set algorithm nor an interior point algorithm, but has characteristics of both. Our algorithm is globally convergent and asymptotically 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 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. We implemented our algorithm and tested it on a sequential computer on variety of dense problems, and we present numerical results which show that our algorithm solves many problems quickly.

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 Section 2, we describe and motivate our algorithm. We prove global convergence in Section 3, and superlinear convergence in Section 4. Section 5 contains numerical results and a discussion of the behavior of the algorithm. Finally, in Section 6, we discuss possible improvements and make some concluding remarks.

1.1 A related problem

We will assume that our problem is of the form

$$\begin{array}{c} \min \frac{1}{2}x^T Ax + b^T x \\ -1 \leq x \leq 1, \end{array} \tag{2}$$

where A is symmetric positive definite. Through a suitable scaling and translation we can rewrite (1) in this form. Note that this transformation may adversely affect the conditioning of A, especially if some of the original bounds are large.

Consider the quadratic program (2). 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:

$$\begin{array}{ll} \text{feasibility:} & -1 \leq x^* \leq 1, \\ \text{first order:} & \left\{ \begin{array}{ll} \nabla q_x(x^*)_i = 0 & \text{if } -1 < x_i^* < 1, \\ \nabla q_x(x^*)_i < 0 & \text{if } x_i^* = 1, \\ \nabla q_x(x^*)_i > 0 & \text{if } x_i^* = -1. \end{array} \right. \end{array}$$

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

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

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

$$D\nabla q_x(x^*) = 0.$$

Now consider the following piecewise quadratic minimization problem,

$$\min f(y) = \frac{1}{2}y^T A^{-1} y + y^T A^{-1} b + \|y\|_1. \tag{3}$$

The following conditions are sufficient to guarantee that y^* is a minimum of (3) [2]: there exists a vector λ^* such that

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

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

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

where $Y^* = \operatorname{diag}(y^*)$. Thus if we equate λ^* with x^* and hence y^* with $-\nabla q_x(x^*)$, then we see that a solution to (3), yields a solution to (2). This new formulation gives us a new perspective from which to approach solving (2). 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. 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 $\operatorname{diag}(v)$ 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 $|v_{ij}|$.

To help identify which components have y_i converging to zero and which have $|x_i|$ converging to unity, at each point y, define the set $B(y) = \{i \mid |y_i| \leq \|A^{-1}\|_\infty^{-1} |\, 1 - |x_i|\, |\}$ and its complement $B^c(y)$. The factor $\|A^{-1}\|_\infty^{-1}$ reflects the accuracy with which we can compute x from y. We use these sets in our algorithm to determine when we are sufficiently close the optimal solution. A vector subscripted with one of these sets, e.g., $y_{B(y)}$, denotes only those components of the vector with indices in the subscript set.

For any point y with $y_i \neq 0$ for all i, $\nabla f(y)$ is defined and $\nabla f(y) = A^{-1}y + A^{-1}b + \operatorname{sign}(y)$. 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. Define $S(\alpha, y; s)$ to be the set of breakpoints that the line search has crossed when it reaches α, i.e., $S(\alpha, y, s) = \{i \mid 0 < -y_i/s_i \leq \alpha\}$. Define $\sigma(\alpha, y, s) = \operatorname{sign}(y + \alpha s)$. 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., $\sigma = sign(y)$.

We call a quadratic program of the form in (2) *nondegenerate* on any closed and bounded region C if, for any $x \in C$ with $|x_i| = 1$ for some i, $\nabla q_x(x)_i \neq 0$. If a quadratic program is nondegenerate, the properties of ∇q_x and C insure that there exist $\epsilon > 0$ and $\delta \in (0,1)$ such that if $|x_i| = 1$ for some i, then $|\nabla q_x(x)_i| > \epsilon$, and if $|\nabla q_x(x)_i| = 0$ for some i, then $|\ 1 - |x_i|\ | > \delta$. We will refer to a problem as being near x-degenerate if for some i, $y_i = 0$ and $1 - |x_i|$ is "small", and near y-degenerate if for some i, $|x_i| = 1$ and $|y_i|$ is "small". On a given problem, in order to guarantee that our algorithm converges to a feasible point and that it converges superlinearly, we need to make a *nondegeneracy assumption*, i.e., we assume that the problem is nondegenerate for all points in a closed domain that we will specify later.

2 The algorithm

In this section, we motivate and describe our algorithm for solving (3). From equations (4), we see that a solution to (3) is also a zero of

$$F(y) = Y(A^{-1}y + A^{-1}b + \text{sign}(y)) = 0. \tag{5}$$

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

$$s_N = -(YA^{-1} + \text{diag}(\nabla f(y)))^{-1}Y\nabla f(y). \tag{6}$$

The following lemma shows that in a neighborhood of the solution of (3), the Newton step is actually a descent direction for $f(y)$.

Lemma 2.1 *Under the nondegeneracy assumption, 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 (6) as

$$s_N = -(A^{-1} + Y^{-1}\text{diag}(\nabla f(y)))^{-1}\nabla f(y), \tag{7}$$

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 $-s_N^T\nabla f(y) > 0$. The idea behind the proof is that if $i \in B^c(y^*)$, then $\nabla f(y)_i/y_i$ converges to 0, hence in a neighborhood of y^*, $|\nabla f(y)_i|/|y_i|$ is dominated by $(1/\|A\|_2)$. If $i \in B(y^*)$, then $\nabla f(y)_i/y_i$ converges to $+\infty$, hence in a neighborhood of y^*, $\nabla f(y)_i/y_i > 0 > -(1/\|A\|_2)$.

Set

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

where

$$ydg = \min_{i \in B^c(y^*)} |y_i^*|, \text{ and } xdg = \min_{i \in B(y^*)} (1 - |x_i^*|).$$

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

If $i \in B^c(y^*)$ 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 $\nabla f(y)_i / y_i > -(1/\|A\|_2)$ for $i \in B^c(y^*)$.

If $i \in B(y^*)$ 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. ■

To devise an algorithm that is globally convergent, we would like a step that is always a descent direction for the function f. To accomplish this, we choose a step of the form

$$s = -(A^{-1} + Y^{-1}\hat{R})^{-1} \nabla f(y), \tag{8}$$

where $\hat{R}$ is a diagonal matrix with the property that $\hat{r}_{ii}\sigma_i > 0$ for all i. If we define $R = \Sigma \cdot \hat{R}$, where $\Sigma = \text{diag}(\sigma_i)$, we can rewrite (8) as

$$s = -(|Y|A^{-1} + R)^{-1} |Y| \nabla f(y), \tag{9}$$

where now we require $r_{ii} \geq 0$ for all i. Thus we have the following lemma.

Lemma 2.2 *The search direction s is a descent direction, i.e., $-s^T \nabla f(y) > 0$.*

In order for s to approach the Newton step, R must approach $\Sigma \cdot \text{diag}(\nabla f(y))$. A natural choice of R would be $\text{diag}(|\nabla f(y)|)$. Such a step would also be a descent direction for $q_x(x)$ if x were feasible. However, recall that it is not sufficient to satisfy

$Y(A^{-1}y + A^{-1}b + \text{sign}(y)) = 0$; we also need to insure that λ (i.e., x) is feasible. We choose $R = \text{diag}(r)$, where

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

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

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

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 does approach $\Sigma \cdot \text{diag}(\nabla f(y))$. We can prove the following useful lemma about R.

Lemma 2.3 *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 $0 \leq \theta \leq 1$, each term is greater than or equal to zero. Thus $r_i = 0$ if and only if $\theta = 0$ and $\nabla f(y)_i = 0$. ∎

Once we have computed our search direction s, we determine the step length α by a piecewise quadratic line search. Before describing this procedure, we introduce some notation and describe the geometry of the line search. 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 consecutive breakpoints, say β_i and β_j, $f_{y,s}(\nu)$ is a quadratic. 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 quadratics 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 slope is the same for all intervals. For any ν, let β_i and β_j be the two consecutive breakpoints surrounding ν, and define $\gamma(\nu)$ to be the step from ν to the minimum of $f_{(i,j)}(\nu)$, i.e.,

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

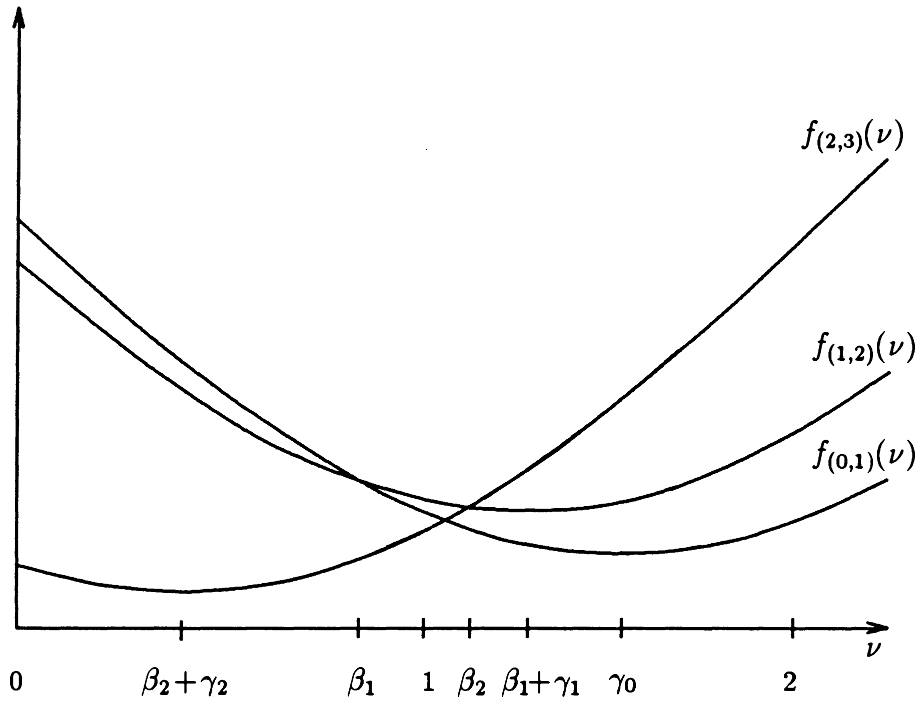

Figure 1: The quadratic functions that comprise the piecewise quadratic $f_{y,s}(\nu)$.

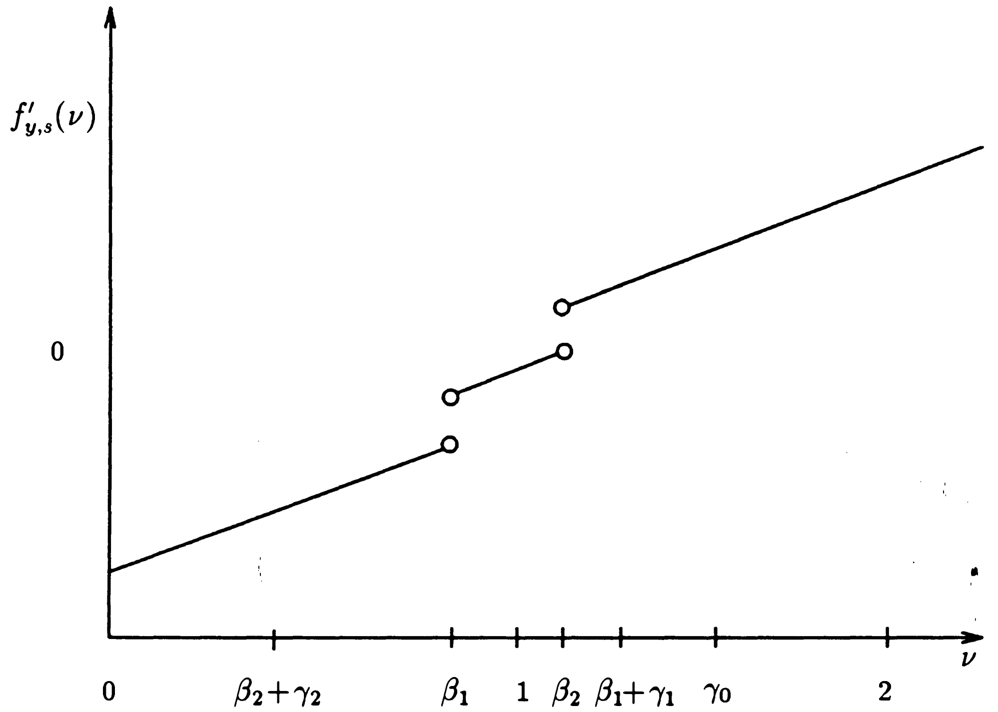

Figure 2: The function $f'_{y,s}(\nu)$

For notational convenience, define $\beta_0 = 0$ and $\gamma_i = \gamma(\beta_i)$. Figures 1 and 2 illustrate these quantities.

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

Lemma 2.4 *The functions $-s^T g(\nu, y, s)$ and $\gamma(\nu)$ are monotonically decreasing functions of ν.*

Proof: We have

$$-s^T g(\nu, y, s) = -s^T g(0) - \nu s^T A^{-1} s - s^T(\sigma(\nu) - \sigma(0)). \tag{10}$$

But

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

So

$$s^T(\sigma(\nu) - \sigma(0)) = \sum_{i \in S(\nu)} (-2\sigma_i s_i) = 2 \sum_{i \in S(\nu)} |s_i| \tag{11}$$

is greater than 0 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. ∎

Now we describe the details of the line search. To determine the step length, 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. 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 $\beta_{p(i+1)}$. In this case, since y must not be a point where f is not differentiable, yet it must approach such a point in the limit, instead of setting $\alpha = \beta_{p(i)}$, set $\alpha = \beta_{p(i-1)} + \tau(\beta_{p(i)} - \beta_{p(i-1)})$, where $\tau = \min(c_2, 1 - \theta/c_1)$, and $0 < c_2 < 1$.

The complete algorithm is outlined in Figure 3.

2.1 Implementation details

In this section, we describe an efficient and numerically stable way to implement our algorithm. 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, we do not want to form A^{-1}. In addition, if A is sparse,

$y = y^0$

$\rho = 1/\|Y\nabla f(y)\|_1$ evaluated at a "typical" value of y

while not optimal **do**

$\lambda = -(A^{-1}y + A^{-1}b)$

$\nabla f(y) = -\lambda + \mathrm{sign}(y)$

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

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

$R = (\theta I + (1-\theta)\mathrm{diag}(|\nabla f(y)|))$

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

determine α by the line search described above

$y = y + \alpha s$

enddo

Figure 3: A superlinear algorithm

generally A^{-1} will not be sparse. Since, by definition, R is non-singular if y is not optimal, we have the following equivalent linear systems 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

$$(|Y| + R^{\frac{1}{2}}AR^{\frac{1}{2}})v = -R^{-\frac{1}{2}}(|Y|\nabla f(y)) \tag{12}$$

for v, then we can easily compute $s = AR^{\frac{1}{2}}v$. Furthermore, this approach is well suited to sparse problems, since the structure of the matrix in (12) 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 [17] 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 break point β_j^k that we cross. From equations (10) and (11), we have

$$-(s^k)^T g^k(\beta_j^k) = -(s^k)^T g^k(0) - \beta_j^k (s^k)^T A^{-1}(s^k) + \sum_{i\in S^k(\beta_j^k)} (2\sigma_i^k s_i^k). \tag{13}$$

(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 an inner product.

Finally, we discuss the stopping criteria for our algorithm. We need to insure that we are "close enough" to satisfying both the feasibility condition and the complementary slackness condition of equation (4). Suppose our computed y satisfies $y = y^* + e$ for some vector e, where y^* is the true solution. Then $\lambda = A^{-1}(y - b) = A^{-1}(y^* - b) + A^{-1}e = \lambda^* + A^{-1}e$. Thus if we expect the components of $y_{B(y)}$ to satisfy $|y_i| < yscale \cdot tol$, where tol is a user specified tolerance, we can expect the components of $\nabla f(y)_{B^c(y)}$ to satisfy $|\nabla f(y)_i| < \|A^{-1}\|_\infty \cdot yscale \cdot tol$. We can approximate $\|A^{-1}\|_\infty$ with $(\kappa_\infty(A)/\|A\|_\infty)$, where $\kappa(A)_\infty$ is an approximation to the infinity norm condition number of A. Then we set $yscale$ to be the magnitude of a "typical" value of y, e.g., $\|A \cdot \text{sign}(-b) + b\|_\infty$, and $gscale$ to be $(\kappa_\infty(A)/\|A\|_\infty) \cdot yscale$. Finally, we define the vector of infeasibility

$$feas = \max\left(|\lambda| - 1, 0\right)$$

and we terminate the algorithm when

$$\max\left(\frac{\|feas\|_\infty}{gscale}, \frac{\|y_{B(y)}\|_\infty}{yscale}, \frac{\|\nabla f(y)_{B^c(y)}\|_\infty}{gscale}\right) < tol.$$

In our implementation, we choose tol to be machine epsilon.

3 Global Convergence

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

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

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

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. We let the domain be

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

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 converges, and then that the step length 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,

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

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

$$\begin{aligned} & f(y^k) - f(y^{k+1}) \\ &\quad = f(y^k) - f(y^k + \alpha^k s^k) \\ &\quad = -(\alpha^k s^k)^T\nabla f(y^k + \alpha^k s^k) + \frac{1}{2}(\alpha^k s^k)^T A^{-1}(\alpha^k s^k) + (\sigma^k(0) - \sigma^{k+1}(0))^T y^k. \end{aligned}$$

But

$$\begin{aligned} (\sigma^k(0) - \sigma^{k+1}(0))^T y^k &= (\sigma^k(0) - \sigma^k(\alpha))^T y^k \\ &= 2\sum_{i\in S^k(\alpha)} |y_i^k|. \end{aligned}$$

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

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

we have

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

Since A^{-1} is positive definite, and $\{f(y^k)\}$ converges, 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 $\hat{\alpha}^k = \min(\gamma_0, c_2\beta^k_{p(1)})$. 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 0. 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. ∎

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.*

Proof: Since $\{y^k\}$ is bounded, it has at least one limit point. Suppose that $\{y^k\}$ has at least two distinct limit points. Let $\hat{y}$ and $\bar{y}$ be the two distinct limit points that are closest together and let $\|\hat{y} - \bar{y}\|_2 = \delta$. Hence $\delta > 0$. We want to derive a contradiction by showing that there must be an limit point closer to $\hat{y}$ than $\bar{y}$, and thereby conclude that there can be only one limit point and so the sequence must converge. Since $\|y^{k+1} - y^k\|_2 \to 0$, we can find K_1 such that for $k > K_1$, $\|y^{k+1} - y^k\|_2 < \frac{\delta}{8}$. Consider the set $C = \{y \mid \frac{\delta}{4} \le \|y - \hat{y}\|_2 \le \frac{\delta}{2}\}$. Suppose C contains at most finitely many points of $\{y^k\}$. If K_2 is the largest numbered of these points, then for $k > K_2$, either $\|y^k - \hat{y}\|_2 < \frac{\delta}{4}$ or $\|y^k - \hat{y}\|_2 > \frac{\delta}{2}$. Let $j > \max\{K_1, K_2\}$. Suppose that $\|y^j - \hat{y}\|_2 < \frac{\delta}{4}$. Then $\|y^{j+1} - \hat{y}\|_2 \le \|y^{j+1} - y^j\|_2 + \|y^j - \hat{y}\|_2 < \frac{3\delta}{8}$. But since $j + 1 > K_2$, then $y^{j+1} \notin C$, so $\|y^{j+1} - \hat{y}\|_2 < \frac{\delta}{4}$. Hence by induction, $\|y^{j+m} - \hat{y}\|_2 < \frac{\delta}{4}$ for all $m \ge 0$. But this contradicts the fact that $\bar{y}$ is an limit point of $\{y^k\}$ such that $\|\hat{y} - \bar{y}\|_2 = \delta$. We can reach a similar contradiction by assuming that $\|y^j - \hat{y}\|_2 > \frac{\delta}{2}$.

Hence there must be infinitely many points of the sequence $\{y^k\}$ in C, so since C is compact, they must have an limit point $\tilde{y}$ in C. But since $\tilde{y} \in C$, we have $\|\tilde{y} - \hat{y}\|_2 \le \frac{\delta}{2}$. Thus we have shown that the sequence $\{y^k\}$ has an limit point closer to $\hat{y}$ than $\bar{y}$ and have reached the desired contradiction. ∎

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,*

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

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

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

and during the line search, the jth breakpoint will not 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 can not 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, $\text{sign}(\lambda_i^k) = -\sigma_i^k$. Thus $\mu_i^k = -1$. So the limit point of the sequence $\{\beta_i^k\}$ is

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

■

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,

$$-(s^k)^T g^k(\beta_j^k) > 0. \tag{14}$$

Proof: From equation (13), we have

$$-(s^k)^T g^k(\beta_j^k) = -(s^k)^T g^k(0) - \beta_j^k (s^k)^T A^{-1}(s^k) + \sum_{i \in S^k(\beta_j^k)} (2\sigma_i^k s_i^k). \tag{15}$$

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 (15) as

$$(1 - \beta_j^k)(s^k)^T A^{-1}(s^k) + (s^k)^T (|Y^k|^{-1} R^k)(s^k) + \sum_{i \in S^k(\beta_j^k)} (2\sigma_i^k s_i^k). \tag{16}$$

For large enough k, Lemma 3.13 shows that $\beta_j^k < 1$, so the first term in (16) is greater than zero. We can express the second and third terms as

$$\sum_{i \in S^k(\beta_j^k)} \left(\frac{r_j^k |s_j^k|}{|y_j^k|} - 2 \right) |s_j^k| \;\; + \sum_{i \notin S^k(\beta_j^k)} \frac{r_i^k}{|y_i^k|} (s_i^k)^2. \tag{17}$$

The second sum in (17) is obviously greater than 0. Thus the only thing remaining to show is that the first sum in (17) is greater than 0. We can simplify the summand in the first sum as follows.

$$\left(\frac{r_i^k |s_i^k|}{|y_i^k|} - 2 \right) |s_i^k| = (|\nabla f(y^k)_i + (A^{-1} s^k)_i| - 2) |s_i^k|. \tag{18}$$

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 (17) is greater than zero, and the proof is complete. ∎

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

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 can not 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

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

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

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

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 it converges to one. 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

$$\{\|W^k - \nabla F_{\nu^k}(y^*)\|\} \to 0, \tag{19}$$

then $\{y^k\}$ 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 (19) 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 equations (10) and (11), we have

$$\gamma^k(\alpha) = \frac{1}{(s^k)^T A^{-1}(s^k)} \left(-(s^k)^T g^k(0) - \alpha (s^k)^T A^{-1}(s^k) - \sum_{i\in S^k(\alpha)} (-2\sigma_i^k s_i^k) \right). \qquad (20)$$

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 (20) 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

$$\begin{aligned}
\gamma^k(\alpha) &= (1-\alpha) + \sum_{i\in S^k(\alpha)} \left(\frac{r_i^k |s_i^k|}{|y_i^k|} - 2 \right) \frac{|s_i^k|}{(s^k)^T A^{-1}(s^k)} + \sum_{i\notin S^k(\alpha)} \frac{r_i^k}{|y_i^k|} \frac{(s_i^k)^2}{(s^k)^T A^{-1}(s^k)} \\
&= (1-\alpha) + \sum_{i\in S^k(\alpha)} (|\nabla f(y^k)_i + (A^{-1}s^k)_i| - 2) \frac{|s_i^k|}{(s^k)^T A^{-1}(s^k)} \\
&\quad + \sum_{i\notin S^k(\alpha)} \frac{r_i^k}{|y_i^k|} \frac{(s_i^k)^2}{(s^k)^T A^{-1}(s^k)}.
\end{aligned}$$

If $i \in S^k(\alpha)$, Lemma 3.17 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.

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

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 $\sum_{i\notin S^k(\alpha)} \frac{r_i^k}{|y_i^k|}$ 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 $\{\beta^k_{p(1)}\} \to 1$ or $\{\beta^k_{p(1)}\} \to \infty$ and Corollary 3.16 implies that $\{\theta^k\} \to 0$. By definition, $\alpha^k \geq (1 - \theta^k/c_2)\beta^k_{p(1)}$, 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 (19) 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 $G_{\sigma^k}(y^k) = \mathrm{diag}(\nabla f(y^k))$. From the definitions of W^k, $\nabla F_{\sigma^k}(y^*)$, and R^k, we have

$$\begin{aligned}
\|W^k - \nabla F_{\sigma^k}(y^*)\|_2 &= \| \frac{1}{\alpha^k}(Y^k A^{-1} + \Sigma^k R^k) - (Y^* A^{-1} + G_{\sigma^k}(y^*))\|_2 \\
&\leq \|(\left(\frac{1}{\alpha^k}\right) Y^k - Y^*)A^{-1}\|_2 + \| \frac{1}{\alpha^k}(\Sigma^k \Gamma^k \; G_{\sigma^k}(y^k) \;) - G_{\sigma^k}(y^*) \; \|_2 \\
&\quad + \frac{\theta^k}{\alpha^k} \; \|\Sigma^k \; (I - |G_{\sigma^k}(y^k)| \;)\|_2,
\end{aligned}$$

where $\Gamma^k = \mathrm{diag}(\mathrm{sign}(\nabla f(y^k)))$. Theorem 3.5 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

$$\| \frac{1}{\alpha^k}(\Sigma^k \Gamma^k G_{\sigma^k}(y^k) \;) - G_{\sigma^k}(y^*) \; \|_2 \to 0 \tag{21}$$

to obtain the desired result. But (21) 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^*.

We have not been able to determine if our algorithm converges quadratically. However, it is easy to see what needs to be proved. Under the assumptions of Theorem 4.1, if

$$\|W^k - \nabla F_{\nu^k}(y^*)\| = O(\|y^k - y^*\|), \tag{22}$$

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 (2) in a manner similar to that 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^*. Our problems differ from theirs only in the choice of $ymag$, as we describe below.

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 $x_i^* = 0$ 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 bound at the solution, generate x^* as follows. First, choose $B^c(y^*)$, a subset of variables to be bound at the solution, by generating a random number μ_i in $(0, 1)$ for each $i = 1, \ldots, n$ and including i in $B^c(y^*)$ if $\mu_i < nb/n$. Then choose x^* by setting those components in $B^c(y^*)$ randomly to $+1$ or -1, and selecting the remaining components by randomly generating x_i in $(-1, 1)$.

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

$$y^* = \text{sign}(\mu_i) \times 10^{ymag} + 100\mu_i \times 10^{\nu_i \cdot ymag}. \tag{23}$$

This choice guarantees that the solutions of the test problems are not near y-degenerate. Then by setting $b = y^* - Ax^*$, we have a problem with the desired characteristics.

5.2 Numerical results

In this section, we examine the behavior of our algorithm. We implemented our algorithm in double precision Fortran, using the numerical linear algebra routines from LINPACK [7]. Our experiments were run on a Sun Sparcstation 1 at Cornell's Computational Optimization Project Laboratory.

We implemented the algorithm as described in Section 2. As our starting point, we chose the origin, i.e., $y^0 = 0$. Empirically, we determined that $c_1 = 10^{-3}$ and $c_2 = 0.90$ were reasonable choices of these parameters and used them in our tests. We set tol equal to machine epsilon, namely, 2.22×10^{-16}.

To capture the behavior of the algorithm, we varied one of our problem parameters while keeping the others fixed. We fixed $n = 100$, restricted *lcnd* to the values 0, 3, 6, 9, and 12, and restricted *nb* to the values 10, 50, and 90. We restricted *ymag* to be 1, 3, 6, 9, or 12, where the magnitude of the nonzero components of y increases as *ymag* increases. We ran 10 problems for each set of problem parameters. Tables 1, 2, and 3 contain the average, maximum, and minimum number of iterations required to solve these problems, whose *solutions* are not near x- or y-degenerate.

Our algorithm converged quickly on most of these problems, although on a few of them it did not. On the slower problems, the minimum value of

$$\min_k \left(\min_{i \in B^c(y^*)} |y_i^k| \right)$$

encountered was generally smaller than on the faster problems. Hence on the slower problems, the algorithm encountered a point of near y-degeneracy. In general, it converged faster on problems with smaller *lcnd* and larger magnitude of *ymag*. To demonstrate the algorithm's typical behavior, in Table 4 we provide some relevant quantities at each iteration of a sample run.

Although our algorithm is not designed for degenerate problems, we examined its behavior on some degenerate and near degenerate problems. Interestingly, it worked better on exactly degenerate problems than on near degenerate problems. On exactly degenerate problems, the algorithm behaved as it did on non-degenerate problems with the same problem parameters. On near x-degenerate problems, i.e., those where the magnitude of the components of $x^*_{B(y^*)}$ are near 1, the performance of our algorithm degraded as the problems became more nearly x-degenerate. The algorithm required more iterations when *nb* was smaller, especially on more nearly x-degenerate problems. It seems to have difficulty when there are many components of x that are near one in magnitude. On near y-degenerate problems, i.e., those where the magnitude of the components of $y^*_{B^c(y^*)}$ are small, our algorithm converged quickly on some problems and slowly on others. Even for problems with the same parameters, there was a wide discrepancy among the number of iterations required for convergence. Here again, on the slowly converging problems, the minimum value of

$$\min_k \left(\min_{i \in B^c(y^*)} |y_i^k| \right)$$

encountered was generally smaller than on the faster problems.

Our algorithm apparently has trouble when it encounters near degeneracy. There are several reasons why this might be the case. Examining the proofs of Lemmas 3.11, 3.14, and 4.3 we see that for near degenerate problems, the "for large enough k" might need to be quite large. Additionally, the radius of convergence of Newton's method is likely to be quite small if the solution is near degenerate. Finally, the algorithm can have trouble reaching a feasible point in the near degenerate case. Suppose that for some i, $x_i^* = 1$ and $\nabla q_x(x^*)_i = -\epsilon$, for some small $\epsilon > 0$. It is likely that for some small positive $\hat{\alpha}$, $\nabla q_x(x^* + \hat{\alpha} e_i)_i$ will be zero. Furthermore, $x^* + \hat{\alpha} e_i$ is infeasible, yet close to being feasible. The algorithm seems to be attracted to such points and to have trouble

Table 1: Iterations for $nb = 10$.

	ymag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	11.1	10	12	10.8	10	12	10.6	10	12	10.0	9	12	7.9	6	10
3	11.6	10	13	11.4	10	13	11.3	10	13	10.5	10	12	9.8	9	12
6	14.6	13	17	12.7	11	15	12.3	10	14	11.8	10	14	11.2	10	14
9	16.4	14	20	15.0	13	17	13.8	11	16	13.0	11	15	12.1	10	15
12	16.2	13	21	15.9	13	20	15.2	12	22	14.5	10	24	13.0	10	17

Table 2: Iterations for $nb = 50$.

	ymag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	10.3	9	12	10.1	9	11	9.5	8	11	9.2	7	11	6.7	5	9
3	10.6	10	11	10.4	9	12	10.2	8	12	9.5	7	11	8.6	7	11
6	14.4	14	15	12.7	12	14	11.4	10	13	10.8	10	12	9.7	9	11
9	16.6	14	27	14.3	13	15	13.4	12	16	12.0	10	14	10.8	10	12
12	16.3	12	24	13.8	11	17	15.7	12	19	14.7	12	19	11.8	10	14

Table 3: Iterations for $nb = 90$.

	ymag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	8.2	6	9	7.9	7	9	7.5	6	9	6.9	5	8	5.7	4	7
3	10.6	9	14	8.2	6	10	7.6	6	9	7.3	6	9	6.4	5	8
6	19.5	13	32	15.4	10	24	9.3	8	10	8.3	7	9	7.7	6	9
9	23.2	13	71	18.4	12	43	18.6	11	27	9.8	8	11	8.1	7	9
12	16.7	10	28	17.8	11	28	18.1	12	25	18.7	10	33	10.1	8	12

Table 4: A sample run, $lcnd = 6$, $ymag = 6$, $nb = 50$

it	$\frac{\|y_{B(y)}\|_\infty}{yscale}$	$\frac{\|\nabla f(y)_{B^c(y)}\|_\infty}{gscale}$	$\frac{\|feas\|_\infty}{gscale}$	$\|x - x^*\|_2$	$\|y - y^*\|_2$
1	0.00e+00	0.27e–07	0.00e+00	0.78e+01	0.12e+07
2	0.00e+00	0.32e–07	0.50e–08	0.27e+01	0.25e+06
3	0.13e–07	0.38e–07	0.23e–09	0.17e+01	0.83e+05
4	0.19e–07	0.48e–07	0.85e–10	0.11e+01	0.25e+05
5	0.78e–08	0.37e–07	0.28e–11	0.38e+00	0.51e+04
6	0.13e–07	0.51e–07	0.43e–12	0.10e+00	0.38e+03
7	0.60e–09	0.36e–08	0.12e–14	0.22e–01	0.18e+02
8	0.22e–09	0.23e–16	0.23e–16	0.21e–02	0.20e–01
9	0.32e–11	0.41e–19	0.15e–19	0.28e–04	0.26e–03
10	0.19e–17	0.12e–22	0.12e–22	0.11e–10	0.31e–09

breaking away from them. Further work on the degenerate and near degenerate cases is needed.

6 Conclusions

In this paper, we presented a new globally and superlinearly convergent algorithm for solving convex quadratic programs with simple bounds. The algorithm was derived using a new approach: we minimized a piecewise quadratic function whose minimum is the optimal point of the original quadratic program. We attained superlinear convergence because our search direction approaches the Newton step for finding a zero of the complementary slackness conditions. Since it only requires the Cholesky factorization of systems with the same sparsity structure, our algorithm is suitable for sparse problems. Our numerical experiments showed that our algorithm converges quickly on many problems, but that it can be lead astray when it encounters near degeneracy.

We are working on modifications to our algorithm to improve its performance in the near degenerate case. Some of our ideas include basing the search direction on the Newton step of a function with a zero only at the optimal point and replacing the function $f(y)$ with a penalty function formulation of the quadratic program to determine the step length in the line search. Another idea is to use a hybrid algorithm that begins resembling a slow-but-steady polynomial algorithm and ends resembling our superlinear algorithm. We are also studying the convergence rate of our algorithm to determine if it is quadratic.

Ideas for future work include determining the effect of different starting points on our algorithm's behavior and extending our algorithm to problems with equality constraints in addition to the simple bounds or to problems with an indefinite matrix A. Additionally, a similar algorithm might be useful for solving general convex minimization problems with simple bounds.

7 Acknowledgement

The authors would like to thank Mike Todd for pointing out a flaw in the proof of convergence in a previous draft and for suggesting an improvement to the convergence criteria and Yuying Li for numerous helpful discussions.

References

[1] Åke Björck. A direct method for sparse least squares problems with lower and upper bounds. *Numerische Mathematik*, 54:19–32, 1988.

[2] Thomas F. Coleman and Andrew R. Conn. Second-order conditions for an exact penalty function. *Mathematical Programming*, 19:178–185, 1980.

[3] Thomas F. Coleman and Laurie A. Hulbert. A direct active set algorithm for large sparse quadratic programs with simple bounds. *Mathematical Programming B*, 1990. (To appear).

[4] Thomas F. Coleman and Yuying Li. A global and quadratic affine scaling method for linear l_1 problems. Technical Report 89–1026, Cornell University, 1989.

[5] Ron S. Dembo and Ulrich Tulowitzki. On the minimization of quadratic functions subject to box constraints. Technical Report B 71, Yale University, 1983.

[6] J. E. Dennis, Jr. and J. J. Moré. Quasi-newton methods, motivation and theory. *SIAM Review*, 19:46–89, 1977.

[7] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart. *LINPACK Users' Guide*. SIAM Publications, 1979.

[8] Chi-Geun Han, Panos M. Pardalos, and Yinyu Ye. Computational aspects of an interior point algorithm for quadratic programming problems with box constraints. In Thomas F. Coleman and Yuying Li, editors, *Proceedings of the Workshop on Large-Scale Numerical Optimization*. SIAM, 1990. (To appear).

[9] J. J. Júdice and F. M. Pires. Direct methods for convex quadratic programs subject to box constraints. Departamento de matemática, Universidade de Coimbra, 3000 Coimbra, Portugal, 1989.

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

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

[12] Jorge J. Moré and Gerardo Toraldo. Algorithms for bound constrained quadratic programming problems. *Numerische Mathematik*, 55:377–400, 1989.

[13] D. P. O'Leary. A generalized conjugate gradient algorithm for solving a class of quadratic programming problems. *Linear Algebra and its Applications*, 34:371–399, 1980.

[14] U. Öreborn. *A direct method for sparse nonnegative least squares problems.* PhD thesis, Department of Mathematics, Linköping University, Linköping, Sweden, 1986.

[15] E. K. Yang and J. W. Tolle. A class of methods for solving large convex quadratic programs subject to box constraints. Department of operations research, University of North Carolina, Chapel Hill, North Carolina, 1988.

[16] Yinyu Ye. An extension of Karmarkar's algorithm and the trust region method for quadratic programming. Technical report, Stanford University, 1986.

[17] Yinyu Ye. Interior point algorithms for quadratic programming. Technical Report 89–29, Department of Management Sciences, University of Iowa, 1989.

[18] Yinyu Ye and Edison Tse. An extension of Karmarkar's projective algorithm for convex quadratic programming. *Mathematical Programming*, 44:157–179, 1989.