Centering, Trust Region, Reflective Techniques for Nonlinear Minimization Subject to Bounds*

Yuying Li**

TR 93-1385
September 1993

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

*This paper was written in preparation for the Sixth Leslie Fox Prize presentation in June of 1993 in Oxford, England. A condensed version of the paper appeared in the proceedings of the 1993 Conference on Scientific and Engineering Computing for Chinese Young Scientists.

** 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 in part by NSF, AFOSR, and ONR through grant DMS-8920550, and by the Cornell Theory Center which receives major funding from the National Science Foundation and IBM Corporation, with additional support from New York State and members of its Corporate Research Institute.

Centering, Trust Region, Reflective Techniques for Nonlinear Minimization Subject to Bounds[1]

Yuying Li[2]
Computer Science Department
and
Advanced Computing Research Institute[3]
Cornell University
Ithaca, New York 14853

September 17, 1993

Abstract. Bound-constrained nonlinear minimization problems occur frequently in practice. Most existing methods belong to an active set type which can be slow for large scale problems. Recently, we proposed a new approach [7, 6, 8] which generates iterates within the strictly feasible region. The method in [8] is a trust region type and, unlike the existing trust region method for bound-constrained problems, the conditions for its strong convergence properties are consistent with algorithm implementation. A reflective technique can be included in the method. In this paper, we motivate techniques which are important for our new approach. Numerical experience on some medium size problems is included.

[1] This paper was written in preparation for the Sixth Leslie Fox Prize presentation in June of 1993 in Oxford, England. A condensed version of the paper appeared in the proceedings of the 1993 Conference on Scientific and Engineering Computing for Chinese Young Scientists.

[2] 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 in part by NSF, AFOSR, and ONR through grant DMS-8920550, and by the Cornell Theory Center which receives major funding from the National Science Foundation and IBM Corporation, with additional support from New York State and members of its Corporate Research Institute.

[3] The Advanced Computing Research Institute is a unit of the Cornell Center for Theory and Simulation in Science and Engineering (Theory Center).

1. Introduction. We consider an important class of nonlinear minimization problems: minimization of a smooth nonlinear function subject to bound constraints on the variables. A bound constrained problem can be described as

$$\min_{x \in \Re^n} f(x), \quad l \le x \le u, \tag{1.1}$$

where $l \in \{\Re \cup \{-\infty\}\}^n$, $u \in \{\Re \cup \{\infty\}\}^n$, $l < u$, and $f : \Re^n \to \Re^1$. We denote the feasible set $\mathcal{F} \stackrel{\text{def}}{=} \{x : l \le x \le u\}$ and the strict interior $int(\mathcal{F}) \stackrel{\text{def}}{=} \{x : l < x < u\}$.

Solving this problem involves *two* tasks: one is to efficiently handle bounds imposed on the variables. The other is the *nonlinear* aspect of the problem. The presence of bound restrictions adds a combinatorial nature to the problem and makes this problem more complicated but interesting.

Recently we have developed a new approach which has good computational performance and strong theoretical properties [7, 6, 8]. The purpose of this paper is to discuss important techniques which are essential to the development of our methods. The presentation of the paper is divided into three components. In §2, we first indicate that for large problems, many existing approaches can be inefficient because of the way bounds are handled. By examing a simpler problem, namely a linear program, we indicate possible techniques to achieve better efficiency. Then, in §3, we will review the trust region idea which is an appealing way to handle the nonlinear aspect of a problem. Moreover, we will illustrate the gap, between the convergence theory and implementation, which exists for other trust region type methods (e.g., [9]). Finally, in §4, we motivate and describe in detail our new approach which, we believe, is superior to existing approaches in both regards. Some numerical experience is reported in §5 and conclusion remarks are given in §6.

The following assumption is made throughout the presentation:

(AS.1) Given an initial point $x_0 \in \mathcal{F}$, we assume that $\mathcal{L}$ is compact where $\mathcal{L}$ is the level set, i.e., $\mathcal{L} = \{x : x \in \mathcal{F} \text{ and } f(x) \le f(x_0)\}$.

2. Techniques for Handling Linear Inequalities. There are many special algorithms tailored for minimizing nonlinear functions with bound constraints (e.g., [3, 7, 9, 10, 13, 15, 20]). Most existing methods handle bounds on variables using an active set strategy.

Minimization algorithms typically generate a sequence of points $\{x_k\}$. Each new point x_{k+1} is a step from the current point x_k: $x_{k+1} = x_k + s_k$. The step s_k is usually a good step which can result in sufficient decrease of $f(x)$.

An active set method for (1.1) seeks an improved point $x_{k+1} \in \mathcal{F}$ by keeping the variables currently at their bounds fixed, when possible. At the end of each iteration, a new variable may reaches one of its bounds (and thus becomes a binding variable). If it is not possible to reduce $f(x)$ under the requirement that all the binding variables remain at their bounds, e.g., at a vertex, a decision is made to leave the binding bound of a variable. If this is not possible, a local minimum of (1.1) is found. The combinatorial nature of the active set technique makes it unsuitable for large-scale problems. Although techniques allowing for more rapid change of the active set have been considered for recent active set type methods (e.g.,[4, 9, 10]), the sequential boundary following nature of the active set approach remains. Therefore, the number of iterations required to solve a problem typically grows as the dimension of the problem (1.1) increases.

In order to see how bound restrictions can be dealt with more efficiently, we temporarily simplify the problem (1.1) by assuming that its objective function is linear. Under this assumption we have a very simple linear programming problem.

Until about a decade ago, the only well-accepted efficient method to solving a general linear programming problem had been the simplex method. The simplex method is essentially an active-set method. In

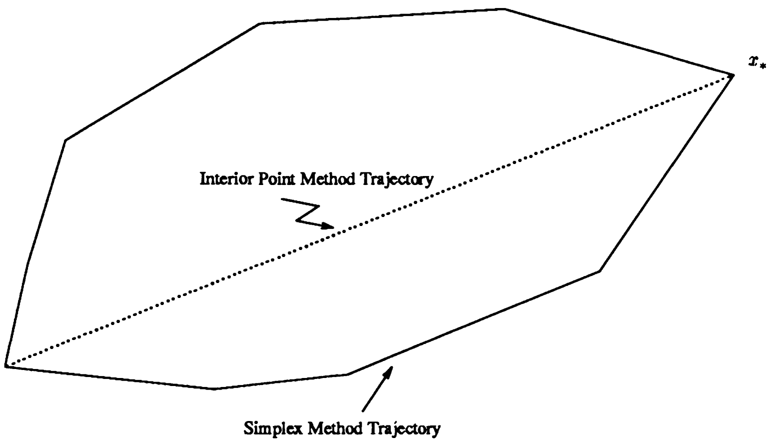

FIG. 1. *Trajectories of the Interior Point Method and Simplex Method*

1984, Karmarkar proposed a projective scaling method which has generated tremendous interest in optimization community. In addition to its polynomial complexity, Karmarkar's projective scaling method has an important and attractive property: it typically requires a small (between 30-70) number of iterations to solve a large problem (tens of thousands of variables). This feature makes the method particularly suitable for large-scale problems.

The simplex method and projective scaling method are very different. While the former is a finite iterative method, the latter an infinite iterative method. Geometrically, the projective scaling method looks distinctively different from the traditional simplex method: a solution is approximated from the points within the center of the feasible region rather than following its boundaries, see FIG. 1.

Karmarkar's projective scaling algorithm has promoted a class of interior point algorithms for linear programming problems. The affine scaling method is an example. In fact, in practice, it is the affine scaling method which has enjoyed the most use. The affine scaling method was proposed and analyzed by Barnes in 1986 [2] and Vanderbei, Meketon and Freeman (1986) [23]. This method is a resurrection of an old method, first suggested and analyzed by Dikin in 1967 [11]. The iterates generated by the affine scaling method stay within the strictly feasible region. Compared to the Karmarkar's projective scaling method, the affine scaling method is simple, intuitive and works on the problem formulation directly. Despite the lack of polynomial complexity [18], an affine scaling method usually performs well in practice [1, 23] and has the similar properties that a small number of iterations is typically needed to solve a large problem.

Since our problem (1.1) is nonconvex and we are not concerned with its complexity issues, we are interested in affine scaling techniques here. We use the box constraint to illustrate the basic idea of the affine scaling method: the centering scheme.

Assume that we have a strictly feasible $x_k \in int(\mathcal{F})$, i.e., $l < x_k < u$.

As indicated in FIG. 2, the sides of the box restrict the movement of the current point x_k. Hence it is reasonable to change units of each variable so that, in the new coordinates, the current feasible point x_k becomes equally distant to all the nearest sides of the box: $\hat{x}_k - D_k^{\text{affine}} \min(x_k - l_k, u_k - x_k) = D_k^{\text{affine}} \min(x_k - l_k, u_k - x_k) = e$ where

$$D_k^{\text{affine}} = \text{diag}(\min(x_k - l_k, u_k - x_k))^{-1}.$$

The benefit of centering is that, in the new coordinates, sufficient progress can be made to reduce the objective function before a bound of a variable is reached.

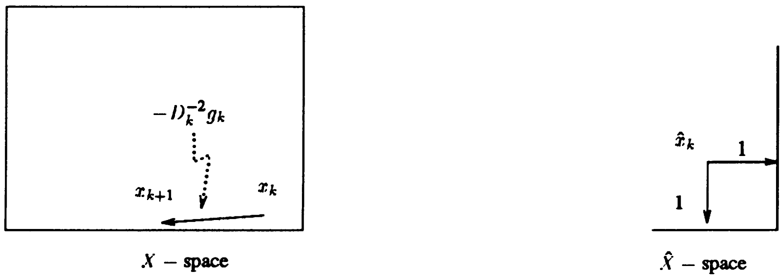

FIG. 2. *Effect of Affine Scaling Transformation* $\hat{x} = D_k^{\text{affine}} x$

In particular, the best direction to take, in the new coordinates, is the steepest descent: $\hat{d}_k = -(D_k^{\text{affine}})^{-1}c$. This corresponds to the scaled steepest direction $d_k = -(D_k^{\text{affine}})^{-2}c$ in the original variable space. This direction is geometrically angled away from the approaching bound, see FIG. 2. Moving along d_k, a step is determined from the current point to the nearest boundary and a large fraction (e.g.,0.9) of this step is taken to stay strictly feasible.

From $d_k = -(D_k^{\text{affine}})^{-2}c$, it is clear that, while the active set methods follow the boundary *exactly* in a *sequential* fashion, the affine scaling method traces the boundary *approximately* but *simultaneously*. This explains why the affine scaling method typically takes fewer iterations to reach the neighborhood of a solution (the total number of iterations is usually relatively insensitive to the problem size). For nonlinear minimization problems, curvature information is a crucial factor in determining a good step. Restricting the iterates to stay on the boundary may fail to produce such a step.

Although the affine scaling method can approach the neighborhood of a solution quickly, it is a *linearly* convergent method.

Our first objectives is to adapt the affine scaling technique to the problem of minimizing a nonlinear function subject to bounds in a manner that quadratic convergence can be achieved.

3. Trust Region Idea. In this section, we concentrate on the other important aspect of a nonlinear bound constrained problem: the nonlinear function $f(x)$. The nonlinear aspect of an optimization problem can be successfully dealt with by the trust region idea.

The trust region idea dates back to Levenberg (1944,[16]) and Marquadt (1963,[17]) for nonlinear least squares problems. Since then, the trust region methods have been developed for unconstrained minimization, linearly constrained nonlinear minimization, nonlinear system of equations, and nondifferentiable minimization.

Trust region methods form a respected class of algorithms for solving unconstrained minimization problems. Their popularity comes from the naturalness of the approach, strong convergence properties and the development of reliable, efficient software.

The motivation of a trust region method for $\min_{x \in \Re^n} f(x)$ is very simple. Following Taylor's theorem, within a sufficiently small neighborhood $\|\bar{D}_k s\| \leq \Delta_k$ (trust region) of the current point x_k, the nonlinear function $f(x_k + s)$ can be well approximated by the quadratic $f(x_k) + \psi_k(s)$ where $\psi_k(s) \stackrel{\text{def}}{=} g_k^T s + \frac{1}{2} s^T H_k s$. Let us express the neighborhood by a bound Δ_k on the step s. Then the original problem $\min_x f(x)$ can be approximated solved by the following trust region subproblem:

$$\min_{s \in \Re^n} \{\psi_k(s) \stackrel{\text{def}}{=} g_k^T s + \frac{1}{2} s^T B_k s \ : \ \|\bar{D}_k s\| \leq \Delta_k\}. \tag{3.1}$$

Here B_k is an approximation to the Hessian matrix $\nabla^2 f(x_k)$, $\bar{D}_k$ is a scaling matrix and Δ_k is a trust region size. The scaling matrix $\bar{D}_k$ is not essential to the basic trust region idea. However, it is important if a problem is badly scaled. The strong convergence properties are achieved when s_k is a good approximation to a global solution of (3.1).

To determine the step s_k completely, we need to specify the trust region size Δ_k. The trust region size Δ_k is updated in a simple manner based on the agreement between the nonlinear function $f(x) - f(x_k)$ and the quadratic approximation $\psi_k(x)$. This agreement is measured by the ratio

$$\rho_k^f \stackrel{\text{def}}{=} (f(x_k + s_k) - f(x_k))/\psi(s_k).$$

An iteration with $\rho_k^f > \mu$ is said to be *successful*. Otherwise, the iteration is *unsuccessful*. The aim of trust region size updating is to force $\rho_k^f > \mu$ and hence ensure sufficient reduction of the objective function. A general trust region scheme for unconstrained minimization of $f(x)$ is described in FIG. 3.

Algorithm 0: Unconstrained Trust Region Method
For $k = 0, 1, \cdots$
1. Compute $f(x_k)$ and the model ψ_k.
2. Define an approximate solution s_k to subproblem (3.1).
3. Compute $\rho_k^f = (f(x_k + s_k) - f(x_k))/\psi(s_k)$.
4. If $\rho_k^f > \mu$ then $x_{k+1} = x_k + s_k$. Otherwise $x_{k+1} = x_k$.
5. Update the scaling matrix $\bar{D}_k$ and Δ_k.

Updating Trust Region Size
Let $0 < \mu < \eta < 1$ and $\gamma_1 < 1 < \gamma_2$ be given
1. If $\rho_k^f \le \mu$ then $\Delta_{k+1} \in (0, \gamma_1 \Delta_k]$.
2. If $\rho_k^f \in (\mu, \eta)$ then $\Delta_{k+1} \in [\gamma_1 \Delta_k, \Delta_k]$.
3. If $\rho_k^f \ge \eta$ then $\Delta_{k+1} \in [\Delta_k, \gamma_2 \Delta_k]$.

FIG. 3. *Trust Region Method for Unconstrained Minimization*

There are a few choices for the norm used to bound a step in the trust region subproblem (3.1). The usual choice is 2-norm. Though it is not easy to compute a global solution to the trust region subproblem with a ball constraint, many techniques are available to compute a sufficiently good approximation [5, 7, 14, 19, 21, 22].

Trust region idea has been used in the context of bound constraints as well. The ∞-norm is a natural choice for a linearly constrained problem, [9, 12]. In [9], the following quadratic programming subproblem is solved:

$$\min_s \{\psi_k(s) : \ \|\hat{D}_k s\|_\infty \le \Delta_k, l \le x_k + s \le u\}.$$

Unfortunately, there is a problem with this norm selection. The strong convergence properties established for trust region type methods hinge upon a good approximation to a global solution of a trust region subproblem. Since the computational methods for a quadratic programming subproblem usually guarantee a local solution at best, a wide gap exists between the convergence theory and implementation of trust region methods for bound constrained problems.

4. An Interior and Trust Region Approach. We have analyzed possible techniques for handling bounds and the nonlinear aspect of (1.1). We have demonstrated some problems associated with the most existing methods:

- possible inefficiency due to sequential boundary following behavior of active set methods for large-scale problems;
- slow local convergence of the affine scaling method for linear programs;
- discrepancy between the theory and implementation of the trust region method type method for (1.1).

We have also illustrated how the first problem can be overcome with the affine scaling technique. Based on the above analysis, we have the following objectives:

- adapt the affine scaling technique to nonlinear minimization with bound constraints;
- achieve *quadratical* convergence;
- use the trust region idea to obtain strong convergence properties which are achievable in implementation.

First, we define a vector function $v(x) : \Re^n \rightarrow \Re^n$ as follows.

DEFINITION 4.1. *The vector $v(x) \in \Re^n$ is defined:*

(i). *If $\nabla f(x)_i < 0$ and $u_i < \infty$ then $v_i \stackrel{\text{def}}{=} x_i - u_i$.*
(ii). *If $\nabla f(x)_i \geq 0$ and $l_i > -\infty$ then $v_i \stackrel{\text{def}}{=} x_i - l_i$.*
(iii). *If $\nabla f(x)_i < 0$ and $u_i = \infty$ then $v_i \stackrel{\text{def}}{=} -1$.*
(iv). *If $\nabla f(x)_i \geq 0$ and $l_i = -\infty$ then $v_i \stackrel{\text{def}}{=} 1$.*

Following **Matlab** notation, for any $s \in \Re^n$, diag(s) denotes an n-by-n diagonal matrix with the vector s defining the diagonal entries in their natural order. In addition, we define

$$D(x) \stackrel{\text{def}}{=} \text{diag}(|v(x)|^{-\frac{1}{2}}) \tag{4.1}$$

i.e., D^{-2} is a diagonal matrix with the i^{th} diagonal component equal to $|v_i(x)|$.

We consider the following diagonal system of nonlinear equations:

$$D(x)^{-2}\nabla f(x) = 0. \tag{4.2}$$

In all subsequent discussions, $H(x) \stackrel{\text{def}}{=} \nabla^2 f(x)$ and $g(x) \stackrel{\text{def}}{=} \nabla f(x)$. Assume that x_* satisfies the equation (4.2). The nonlinear system (4.2) demands that if a component of g_* is positive, the corresponding variable is at its lower bound. If a component of g_* is negative, the corresponding variable is at its upper bound. Hence, (4.2) is the first order necessary conditions for a minimizer of (1.1). Unlike the KKT conditions used in many interior point methods, the system (4.2) does not involve strictly feasible dual variables. System (4.2) is continuous but not everywhere differentiable. Nondifferentiability occurs when $v_i = 0$; we avoid such points by restricting $x_k \in int(\mathcal{F})$.

Assume that $x_k \in int(\mathcal{F})$. A Newton step for (4.2) satisfies

$$\hat{M}_k \hat{d}_k^N = -\hat{g}_k \tag{4.3}$$

where $\hat{d}_k^N$ is a Newton step in the new coordinates under the affine scaling transformation $\hat{x} \stackrel{\text{def}}{=} D_k x$ (hence $\hat{d}_k^N = D_k d_k^N$) and

$$\hat{g}_k \stackrel{\text{def}}{=} D_k^{-1} g_k = \text{diag}(|v_k|^{\frac{1}{2}}) g_k,$$
$$\hat{M}_k \stackrel{\text{def}}{=} D_k^{-1} \nabla f_k D_k^{-1} + \text{diag}(g_k) J_k^v.$$

Here $J^v(x) \in \Re^{n\times n}$ corresponds to the Jacobian of $|v(x)|$. It is easy to see that J^v is a diagonal matrix. For example, if all the components of l and u are finite, $J^v = \mathrm{diag}(\mathrm{sgn}(\nabla f))$. If variable x_i has a finite lower bound and an infinite upper bound (or vice-versa) and $\nabla f_i = 0$, we define $J^v_{ii} = 0$ at such a point.

The following lemma can be easily proved.

LEMMA 1. *Assume that $x_* \in \mathcal{F}$.*
(a) If x_* is a local minimizer of (1.1) then $\hat{g}_* = 0$.
(b) If x_* is a local minimizer then $\hat{M}_*$ is positive semi-definite and $\hat{g}_* = 0$.
(c) If $\hat{M}_*$ is positive definite and $\hat{g}_* = 0$ then x_* is a local minimizer of (1.1).

The results of Lemma 1 indicate that computing a local minimizer of a bound constrained problem (1.1) is the same as locating a point such that $\hat{g}_* = 0$ and $\hat{M}_*$ positive semidefinite. Therefore, loosely speaking, we have transformed a bound-constrained problem (1.1) to a problem of finding a local minimizer for some unconstrained problem.

Though $\hat{g}$ and $\hat{M}$ do not correspond to the gradient and Hessian of a specific nonlinear function, Lemma 1 implies that a solution of the following trust region subproblem is a reasonable step

$$\min_{\hat{s}}\{\hat{\psi}_k(\hat{s}) : \|\hat{s}\|_2 \le \Delta_k\}, \tag{4.4}$$

where

$$\hat{\psi}_k(\hat{s}) \stackrel{\text{def}}{=} \hat{g}_k^T\hat{s} + \frac{1}{2}\hat{s}^T\hat{M}_k\hat{s}.$$

Let $s = D_k^{-1}\hat{s}$. Subproblem (4.4) is equivalent to the following problem in the original variable space:

$$\min_{s}\{\psi_k(s) : \|D_k s\|_2 \le \Delta_k\}, \tag{4.5}$$

where

$$\begin{aligned}
\psi_k(s) &\stackrel{\text{def}}{=} s^T g_k + \frac{1}{2}s^T M_k s,\\
C_k &\stackrel{\text{def}}{=} D_k \mathrm{diag}(g_k) J_k^v D_k,\\
M_k &\stackrel{\text{def}}{=} \nabla^2 f_k + C_k.
\end{aligned}$$

It is clear that C_k is a positive semi-definite diagonal matrix. This matrix contains the constraint information. Moreover, in the neighborhood of a local minimizer, the Newton step to (4.2) is a solution to the trust region subproblem (4.5) if the trust region size Δ_k is sufficiently large.

For the subsequent discussion, we use the notation: $p_k \stackrel{\text{def}}{=} \mathrm{argmin}_{s\in\Re^n}\{\psi_k(s) : \|D_k s\| \le \Delta_k\}$.

We have derived a trust region subproblem (4.5) to compute a good step. Now we discuss how to choose the trust region size.

Since our quadratic model $\psi_k(s)$ includes the constraint information and is not an approximation to $f(x) - f(x_k)$, we need to extend the definition of ρ_k^f. By observing that $\psi_k(s)$ is an approximation to $f(x_k+s) - f(x_k) + \frac{1}{2}s^T C_k s$, we extend ρ_k^f as follows

$$\rho_k^f \stackrel{\text{def}}{=} \frac{f(x_k+s_k) - f(x_k) + \frac{1}{2}s_k^T C(x_k)s_k}{\psi_k(s_k)}.$$

Similar to unconstrained trust region methods, ρ_k^f measures the agreement between the nonlinear function $f(x)$ and its quadratic approximation. If $\rho_k^f > \mu$ for some $\mu > 0$, the nonlinear function is reduced by at least a fraction μ of that of the quadratic model. In this case, we say a step is *successful*. Otherwise a step is unsuccessful.

Algorithm 1: An Interior Trust Region Method for Bound Constrained Problems
$x_0 \in int(\mathcal{F})$
For $k = 0, 1, \cdots$

1. Compute f_k, g_k, H_k, and C_k; Define the quadratic model $\psi_k(s) = g_k^T s + \frac{1}{2}s^T(H_k + C_k)s$.
2. Compute s_k an approximate solution based on (4.5) with $x_k + s_k \in int(\mathcal{F})$.
3. Compute

$$\rho_k^f = \frac{f(x_k + s_k) - f(x_k) + \frac{1}{2}s_k^T C_k s_k}{\psi_k(s_k)}.$$

4. If $\rho_k^f > \mu$ then set $x_{k+1} = x_k + s_k$. Otherwise set $x_{k+1} = x_k$.
5. Update the scaling matrix D_k and Δ_k as specified.

Updating Trust Region Size Δ_k
Let $0 < \mu < \eta < 1$, $\gamma_1 < 1 < \gamma_2$ and $0 < \Lambda_l$ be given

1. If $\rho_k^f \leq \mu$ then set $\Delta_{k+1} \in (0, \gamma_1\Delta_k]$.
2. If $\rho_k^f \in (\mu, \eta)$ then set $\Delta_{k+1} \in [\gamma_1\Delta_k, \Delta_k]$.
3. If $\rho_k^f \geq \eta$ then
   set $\Delta_{k+1} \in [\Delta_k, \gamma_2\Delta_k]$.

FIG. 4. *An Interior Trust Region Method for Minimization Subject to Bounds*

In FIG. 4, we describe our trust region method. Comparing Algorithm 1 with Algorithm 0, the descriptions of the two algorithms are almost the same except the definition of ρ_k^f, D_k and the requirement $x_k + s_k \in int(\mathcal{F})$. The requirement that $x_k + s_k \in int(\mathcal{F})$ is necessary because we want approximations $\{x_k\}$ to approach a solution from $int(\mathcal{F})$. Our new algorithm looks simple and elegant for a fairly complicated problem. But will this algorithm compute a solution? To answer this question, we need to demonstrate a step s_k, which satisfies the requirement $x_k + s_k \in int(\mathcal{F})$ and produces a sufficient decrease of $\psi_k(s)$, can be obtained based on the subproblem (4.5).

We answer this question next.

Affine Scaling Transformation $\hat{x} = D_k x$ and Reflection. In order to see why a reasonable step s_k, $x_k + s_k \in int(\mathcal{F})$ can be obtained subproblem (4.5), let us examine the properties of the global solutions $\{p_k\}$ to (4.5).

We analyze the ith component of p_k as an example.

Assume that $|v_{k_i}| = \min(x_{k_i} - l, u - x_{k_i})$. The optimality conditions say that, if x_i is at a boundary, $v_i = 0$. This implies that, locally speaking, x_k is approaching the 'correct' bound according to the prediction of the gradient. In this case, $(D_k))_{ii} = ((D_k^{\text{affine}})^{\frac{1}{2}})_{ii}$. Since p_k is a global solution of (4.5), p_k satisfies

$$(\hat{M}_k + \lambda_k)D_k^{-1}p_k = -D_k^{-1}g_k.$$

Written in another form

$$(D_k^{-1}C_kD_k^{-1} + \lambda_kI)p_k = -D_k^{-2}(g_k + H_kp_k).$$

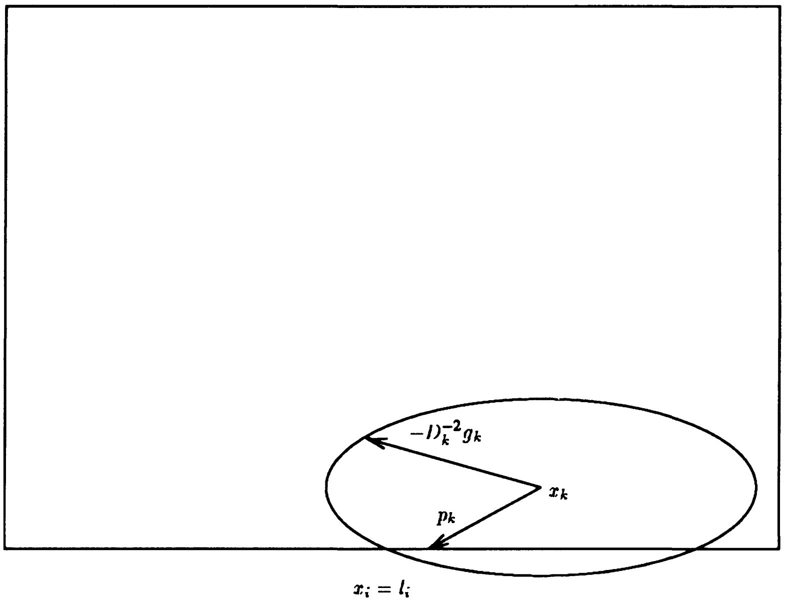

FIG. 5. *Centering*

It is easy to verify from the above equation that $\{(D_k^2)_{ii} p_{k_i}\}$ is bounded under the assumption that $\Delta_k \le \Delta_u$ for some $\Delta_u > 0$. Hence, in the ith component, the effect of the scaling matrix D_k is similar to that of D_k^{affine}. Therefore sufficient progress can be made along p_k in the ith component, e.g., FIG. 5.

Now suppose that $|v_{k_i}| \ne \min(x_{k_i} - l_i, u_i - x_{k_i})$. Locally, this implies that x_{k_i} is approaching an 'incorrect' bound according to g_{k_i}. From the definition of D_k, the shape of the trust region allows a large increment in the variable x_i. Moreover, the scaled steepest descent $-D_k^{-2} g_k$ usually directs away from the bound of x_i in a sharp angle (unless the corresponding gradient is small which is a near degenerate case). Since p_k and $-D_k^{-2} g_k$ has an acute angle and, when the trust region size converges to zero, the trust region solution converges to the scaled steepest descent direction $-D_k^{-2} g_k$, the trust region solution p_k will likely be pointing away from the closest bound of x_i as well. FIG. 6 displays this situation. Again, the bound of x_i will not prevent a sufficient decrease of $\psi_k(s)$.

Nonetheless, it is possible, that p_k points directly towards the closest bound of x_i as illustrated in FIG. 7. From $|v_{k_i}| \ne \min(x_{k_i} - l_i, u_i - x_{k_i})$ and definition of v_k, $v_{k_i} d_{k_i} > 0$. But $g_{k_i} v_{k_i} > 0$ always. Hence we have

$$g_{k_i} d_{k_i} > 0. \tag{4.6}$$

The above inequality is important because it implies that, if only a small step is possible following p_k, the quadratic function $\psi_k(s)$ can be further decreased by following the reflection p_k^R of p_k, i.e., $p_k^R = p_k$ except ${p_{k_i}}^R = -p_{k_i}$. The reflection technique is important under this circumstances because it guides the iterations towards the center of the feasible region and leaves the 'incorrect' bound (see FIG. 7).

Our affine scaling transformation $\hat{x} = D_k x$ differs from the affine scaling transformation used for linear programming problems in two regards: D_k is based on the current gradient g_k and a diagonal component is the *squareroot* of the distance of the corresponding variable to its closest bound, if this bound is correct according to g_k. Our choice of D_k comes naturally from the Newton step of the nonlinear systems characterizing the first order optimality conditions of (1.1). Therefore a global step blends automatically into a Newton step locally and achieves fast local convergence. The advantage of defining the affine scaling transformation based on the gradient g_k is also indicated by FIG. 6 and FIG. 7. The shape

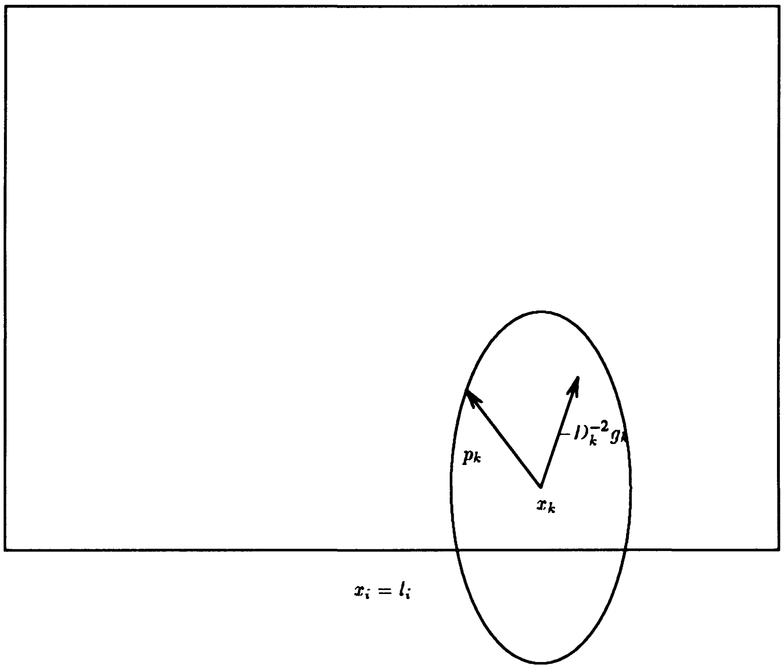

FIG. 6. *Leave an Incorrect Bound*

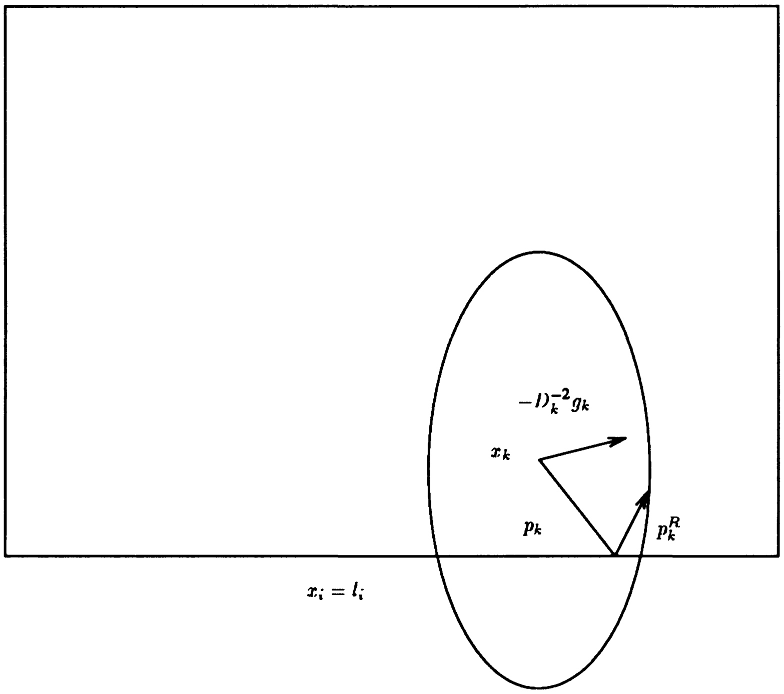

FIG. 7. *Reflection*

of trust region allows a step to leave the 'incorrect' bound based on the current gradient and a reflection technique can be incorporated to achieve this if necessary. For a nonlinear minimization problem, a starting point far away from the center of the feasible region is often provided. Hence it is important for an interior approach to quickly leave the 'incorrect' constraint when applied to a nonlinear problem. This suggests that the scaling matrix $\mathrm{diag}(|v_k|)$ is more reasonable than $\mathrm{diag}(\min(x_k - l, u - x_k))$ because the latter forces the iterates to follow the 'incorrect' bound. We have done some computational experiments to investigate the choice between $D_k^{\mathrm{affine}\frac{1}{2}}$ and D_k. Our experience clearly indicates that D_k is the winner.

A reflection path is formally described as follows. We emphasize, however, a couple of reflections are sufficient in practice.

Define the vector[4]

$$BR_k = \max[(l - x_k) \;./\; d_k, \;\; (u - x_k) \;./\; d_k)], \tag{4.7}$$

where the notation " ./ " indicates component-wise division. Component i of vector BR_k records the positive distance form x_k to the breakpoint corresponding to variable x_{k_i} in the direction d_k. The reflection path is defined in FIG. 8. Since only a single outer iteration is considered, we do not include the subscript k with the variables in our description - dependence on k is assumed.

Definition: Reflection Path $\rho(d)$

[Let $\beta^0 = 0$, $d^1 = d$, set $b^0 = x_k$.]

For $i = 1, 2, \cdots, \infty$

1. Let β^i be the distance to the nearest breakpoint along d^i:

$$\beta^i = \min\{BR : \; BR > 0\}$$

2. Define i^{th} breakpoint: $b^i = b^{i-1} + (\beta^i - \beta^{i-1})d^i$.
3. Reflect to get new dir'n and update BR:
   (a) $d^{i+1} = d^i$
   (b) For each j such that $(b^i)_j = u_j$ (or $(b^i)_j = l_j$)
   - $BR(j) = BR(j) + |\frac{u_j - l_j}{d_j}|$.
   - $(d^{i+1})_j = -(d^i)_j$

FIG. 8. *A Reflection Path* $\rho(d)$

Conditions on a Step s_k. We have discussed why a good step can be obtained based on a solution p_k of (4.5). It is not reasonable, however, to assume that the exact global solution is available, particularly for large scale problems. Instead, approximation to a global solution can be computed. What are conditions s_k need to satisfy in order to guarantee required optimality conditions? We address this issue here.

For simplicity we assume the quadratic model $\psi_k(s) = \nabla f(x_k)^T s + \frac{1}{2} s^T(\nabla^2 f(x_k) + C(x_k))s$. Given any step d_k, a possible step-back may be necessary to stay strictly feasible. We use $\alpha_k^*[d_k]$ to denote the step obtained from d_k with a possible step-back. The exact definition of $\alpha_k^*[d_k]$ is given below. Let τ_k^*

---

[4] For the purpose of computing BR we assume the following rules regarding arithmetic with infinities. If a is a finite scalar then $a + \infty = \infty$, $a - \infty = -\infty$, $\frac{\infty}{a} = \infty \cdot \mathrm{sgn}(a)$, $\frac{-\infty}{a} = -\infty \cdot \mathrm{sgn}(a)$, $\frac{a}{0} = \mathrm{sgn}(a) \cdot \infty$, $\frac{\infty}{0} = \infty$, *and* $\frac{-\infty}{0} = -\infty$, where $\mathrm{sgn}(a) = +1$ if $a \geq 0$, $\mathrm{sgn}(a) < 0$ if $a < 0$.

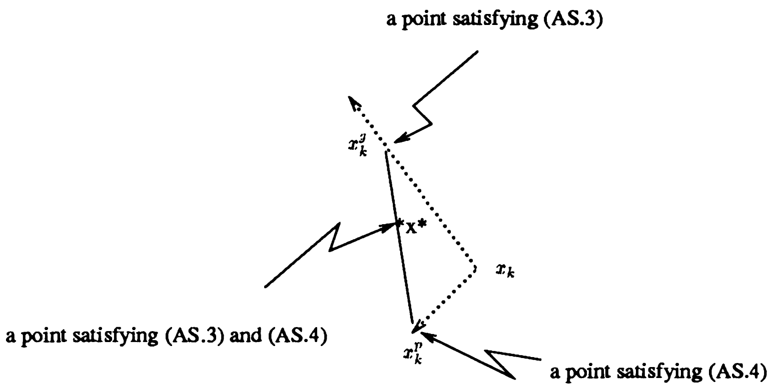

FIG. 9. *An Example: Satisfaction of Both Conditions on Steps*

denote the minimizer along d_k within the feasible trust region, i.e., $\tau_k^* = \text{argmin}_\tau\{\psi_k(\tau d_k) : \|\tau D_k d_k\| \leq \Delta_k, x_k + \tau d_k \in \mathcal{F}\}$, $\theta_k \in [\theta_l, 1]$ for some $0 < \theta_l < 1$ and $\theta_k - 1 = O(\|d_k\|)$. Then

$$\alpha_k^*[d_k] \stackrel{\text{def}}{=} \theta_k \tau_k^* d_k \stackrel{\text{def}}{=} \begin{cases} \tau_k^* d_k & \text{if } x_k + \tau_k^* d_k \in int(\mathcal{F}), \\ \theta_k \tau_k^* d_k & \text{otherwise.} \end{cases} \tag{4.8}$$

The above definition implies that $\theta_k = 1$ if $x_k + \tau_k^* d_k \in int(\mathcal{F})$.

In [8], the convergence results of the trust region methods for (1.1) have been established under the assumptions that a step s_k satisfies the following: given positive constants β, β_0, β^q and β_0^q,

$$\begin{array}{ll} \text{(AS.3)} & \psi_k(s_k) < \beta\psi_k^*[-D_k^{-2}g_k] \\ & \|D_k s_k\| \leq \beta_0\Delta_k,\ x_k + s_k \in int(\mathcal{F}). \\ \text{(AS.4)} & \psi_k(s_k) < \beta^q\psi_k^*[p_k] \\ & \|D_k s_k\| \leq \beta_0^q\Delta_k,\ x_k + s_k \in int(\mathcal{F}). \end{array}$$

The condition (AS.3) is necessary for the first order convergence and (AS.4) is necessary for the second order convergence. Both are extensions of the convergence conditions of unconstrained trust region methods. In particular, when $l = -\infty$ and $u = \infty$, these two assumptions are exactly the same as the ones required by trust region methods for unconstrained problems. However, there is an significant difference: an exact trust region subproblem solution does not necessarily satisfy the condition (AS.3).

Nonetheless, this does not imply that our model problem is inappropriate. As discussed before, a good approximation of a global solution of (4.5) usually satisfies (AS.3). Moreover, there are many ways to compute steps satisfying both conditions. As an example, assuming that x_k^g satisfies (AS.3) and x_k^p satisfies (AS.4), then the minimizer x_k^* of $\psi_k(s)$ along the path connecting the two points satisfies the both conditions, see FIG. 9. In [8], it has been indicated that the trust region size can be updated to help satisfy (AS.3). The reflection technique described in §4 can also be used.

The following convergence properties have been established in [8].

THEOREM 2. *Let the level set $\mathcal{L} = \{x \in \Re^n : f(x) \leq f(x_0),\ x \in \mathcal{F}\}$ be compact and $f : \mathcal{F} \to \Re$ be twice continuously differentiable on $\mathcal{L}$. Let $\{x_k\}$ be the sequence generated by Algorithm 2 under assumption $\psi_k(s) \stackrel{\text{def}}{=} \nabla f(x_k)^T s + \frac{1}{2}s^T(H_k + C_k)s_k$ on the model, and conditions (AS.3) and (AS.4) on the step s_k. Then*

(a) The sequence $\{\hat{g}_k\}$ converges to zero.

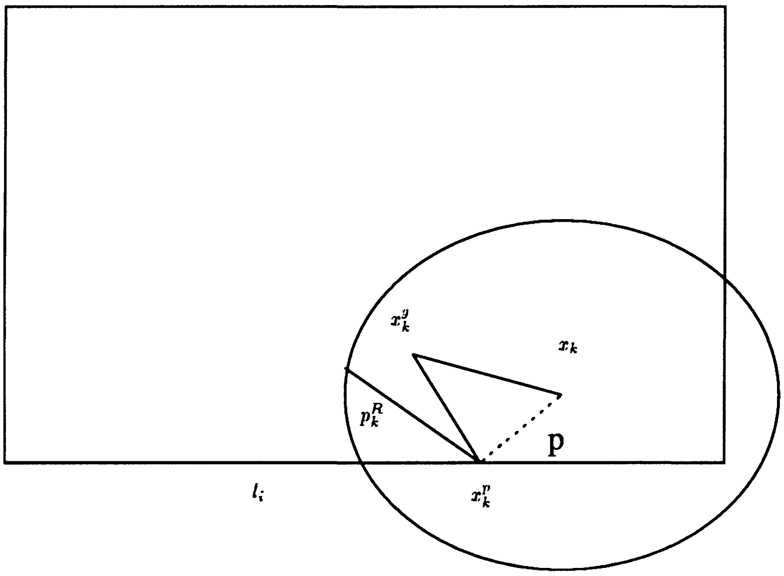

FIG. 10. *Satisfaction of Both Conditions on Steps Through a Reflection-Dogleg Path*

(b) *If every limit point is nondegenerate, then there is a limit point x_* with $\hat{M}_*$ positive semidefinite.*
(c) *If x_* is an isolated nondegenerate limit point, then $\hat{M}_*$ is positive semidefinite.*
(d) *If $\hat{M}_*$ is nonsingular for some limit point x_* of $\{x_k\}$ then*
- *$\hat{M}_*$ is positive definite,*
- *$\{x_k\}$ converges to x_*,*
- *all iterations are eventually successful, and $\{\Delta_k\}$ is bounded away from zero,*
- *$\{x_k\}$ converges to x_* quadratically is $s_k = \alpha_k^*[s_k^N]$ whenever $\|D_k s_k^N\| < \Delta_k$.*

5. Implementation for Dense Problems. In this section, we report some of our experiments with small to medium dense problems. The method implemented is exactly as described as follows.

At each iteration, a step is computed in the following fashion: first a step of (4.5) is computed by solving the full trust region problem. Then a minimum of the quadratic $\psi_k(s)$ is located by following the reflection-dogleg path ρ_k^{RD}: $x_k \to x_k - \alpha_k^*[D_k^2 g_k a] \to x_k + \alpha_k^*[p_k] \to x_k + \alpha_k^*[p_k] + p_k^R$. See FIG. 10.

Though this implementation is not practical for large problems (because it requires a solution to the trust region subproblem), it indicates that the method works and the number of iteration does not seem to increase with the problem size. For the collection of our testing problems, $\alpha_k^*[-D_k^2 g_k]$ is never taken. Unlike the affine scaling method for a linear programming problem, our method is *quadratically* convergent.

The performance of the method for large scale problems (including subspace methods and iterative approaches) is under current investigation. The preliminary results are very encouraging. In [7], numerical results of a line search based method with the same scaling matrix D_k is reported for medium and large scale minimization of general quadratic functions subject to bounds.

6. Conclusion. We have proposed a promising approach for solving large-scale nonlinear bound-constrained problems. By staying within the strictly feasible region, we overcome two difficulties associated with the existing active set methods: the gap between the theory and implementation of the trust region type method for (1.1) and slow changes of active set for large scale problems. In addition,

Number of Function/Gradient Evaluations		
PROB	$n \leq 50$	$n = 100$
GENROSE U	43	23
GENROSE C	12	11
GENSING U	27	22
GENSING C	17	16
CHAINSING U	26	21
CHAINSING C	17	16
DEGENSING U	26	21
DEGENSING C	24	30
GENWOOD U	71	65
GENWOOD C	10	9
CHAINWOOD U	60	84
CHAINWOOD C	10	9
BROYDEN1A U	14	11
BROYDEN1A C	10	9
BROYDEN1B U	7	7
BROYDEN1B C	9	9
BROYDEN2A U	20	19
BROYDEN2A C	20	17
BROYDEN2B U	9	9
BROYDEN2B C	16	15
TOINTBROY U	8	8
TOINTBROY C	10	10
CRAGGLEVY U	33	27
CRAGGLEVY C	28	25
AUGMLAGN U	26	59
AUGMLAGN C	29	31
BROWN1 U	21	20
BROWN1 C	31	30
BROWN3 U	9	9
BROWN3 C	10	9
BVP U	21	15
BVP C	20	16
VAR U	12	9
VAR C	12	23

TABLE 1

Experiments with an Interior, Trust Region and Reflective Method for Bound-constrained Problems our approach has the following important properties.

Firstly, the method is a natural generalization of the trust region method for unconstrained problems. This is possible because of the consideration of the trust region subproblem (4.5). When $l = -\infty$ and $u = \infty$, it becomes a trust region method for unconstrained problem. Hence techniques developed for trust region methods of unconstrained problems are applicable for our method.

Secondly, the iterates generated by the algorithm stay in the strict feasible region and a centering technique is incorporated. The reflection technique makes it easier for the iterates to stay in the strictly feasible region and allows for the second order steps to be taken.

Our computational experience indicates, for the limited testing problems at least, the method works well and has potential for large scale problems.

The basic idea of the approach can be extended to handle additional equality constraints.

REFERENCES

[1] I. Adler, N. Karmarkar, M. Resende, and G. Veiga, *An implementation of karmarkar's algorithm for linear programming*, Math. Prog., 44 (1989).

[2] E. Barnes, *A variation on karmarkar's algorithm for solving linear programming problems*, Mathematical Programming, 36 (1986), pp. 174–182.

[3] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*, Academic Press, London and New York, 1982.

[4] ——, *Projected Newton methods for optimization problems with simple constraints*, SIAM Journal on Control and Optimization, 20 (1982), pp. 221–246.

[5] R. H. Byrd and R. B. Schnabel, *Approximate solution of the trust region problem by minimization over two-dimensional subspaces*, Mathematical Programming, 40 (1988), pp. 247–263.

[6] T. F. Coleman and Y. Li, *On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to boun ds*, Tech. Rep. TR 92-1314, Computer Science Department, Cornell University, 1992.

[7] ——, *A reflective Newton method for minimizing a quadratic funct ion subject to bounds on the variables*, Tech. Rep. TR 92-1315, Computer Science Department, Cornell University, 1992.

[8] ——, *An interior, trust region approach for nonlinear minimization subject to bounds*, Tech. Rep. TR93-1342, Computer Science Department, Cornell University, 1993.

[9] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Mathematics of Computation, 50 (1988), pp. 399–430.

[10] R. S. Dembo and U. Tulowitzki, *On the minimization of quadratic functions subject to box constraints*, Tech. Rep. Working paper series B 71, School of Organization and Management, Yale University, 1983.

[11] I. Dikin, *Iterative solution of problems of linear and quadratic programming*, Doklady Akademiia Nauk SSSR, 174 (1967), pp. 747–748.

[12] R. Fletcher, *An algorithm for solving linearly constrained optimization problems*, mprog, 2 (1972), pp. 133–165.

[13] R. Fletcher and M. P. Jackson, *Minimization of a quadratic function of many variables subject only to lower and upper bounds*, Journal of the Institute for Mathematics and its Applications, 14 (1974), pp. 159–174.

[14] D. M. Gay, *Computing optimal locally constrained steps*, SIAM Journal on Scientific and Statistical Computing, 2 (1981), pp. 186–197.

[15] P. Gill and W. Murray, *Minimization subject to bounds on the variables*, Tech. Rep. Report NAC 71, National Physical Laboratory, England, 1976.

[16] K. Levenberg, *A method for the solution of certain problems in least squares*, Quart. Appl. Math., 2 (1944), pp. 164–168.

[17] D. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters*, SIAM Journal on Applied Mathematics, 11 (1963), pp. 431–441.

[18] N. Megiddo and M. Shub, *Boundary behavior of interior point algorithms in linear programming*, mor, 14 (1989), pp. 97–146.

[19] J. J. Moré and D. Sorensen, *Computing a trust region step*, SIAM Journal on Scientific and Statistical Computing, 4 (1983), pp. 553–572.

[20] J. J. Moré and G. Toraldo, *Algorithms for bound constrained quadratic programming proble ms*, Numerische Mathematik, 55 (1989), pp. 377–400.

[21] G. A. Schultz, R. B. Schnabel, and R. H. Byrd, *A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties*, SIAM Journal on Numerical Analysis, 22(1) (1985), pp. 47–67.

[22] D. Sorensen, *Trust region methods for unconstrained optimization*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 409–426.

[23] R. J. Vanderbei, M. S. Meketon, and B. A. Freedman, *A modification of Karmarkar's linear programming algorithm*, Algorithmica, 1 (1986), pp. 395–407.