Computational Optimization and Applications, 21, 177–199, 2002

A New Trust-Region Algorithm for Equality Constrained Optimization

THOMAS F. COLEMAN
Computer Science Department and Center for Applied Mathematics, Cornell University, Ithaca, NY 14850, USA

JIANGUO LIU
Department of Mathematics, University of North Texas, Denton, TX 76203, USA

WEI YUAN*
Center for Applied Mathematics, Cornell University, Ithaca, NY 14853, USA

Received May 27, 1999; Revised February 16, 2001

Abstract. We present a new trust-region algorithm for solving nonlinear equality constrained optimization problems. Quadratic penalty functions are employed to obtain global convergence. At each iteration a local change of variables is performed to improve the ability of the algorithm to follow the constraint level set. Under certain assumptions we prove that this algorithm globally converges to a point satisfying the second-order necessary optimality conditions. Results of preliminary numerical experiments are reported.

Keywords: nonlinearly constrained optimization, equality constraints, quadratic penalty function, trust-region method, SQP method

1. Introduction

We consider the equality constrained optimization problem

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad c(x) = 0 \end{aligned} \tag{1}$$

where $x \in \Re^n$; $f : \Re^n \to \Re$ and $c : \Re^n \to \Re^m$ are general nonlinear functions. It is assumed that f and c have continuous second-order derivatives.

We propose a new algorithm for (1.1) based, in part, on trust-region techniques. Trust-region methods have proved to be very successful for unconstrained optimization problems, e.g., Moré and Sorensen [23]. Other classical references on this topic are Fletcher [15], Gay [16], Powell [25], and Schultz et al. [27]. Trust-region methods have also been applied to equality constrained problems (e.g., Vardi [28], Byrd, Schnabel, and Schultz [4], Celis et al. [6], and Liu and Yuan [21]), and to general nonlinear programming problems (see, e.g., Boggs et al. [1], Conn et al. [13], and the references therein). In the application of trust

*Present Address: Wireless Systems Engineering, Nortel Technology, Richardson, TX 75083

region methods to equality constrained problems there are two fundamental issues need to be addressed: the specific definition of the trust-region subproblem and the choice of the merit function. A natural way to apply trust-region ideas to problem (1.1) is to consider the trust-region to the SQP subproblem, i.e., minimize a quadratic approximation to the Lagrangian function subject to a linearization of the equality constraint and a trust-region constraint at each iteration. More precisely, the following trust-region subproblem can be solved for the search direction at an iterate x_k:

$$\begin{aligned} \text{minimize} \quad & \nabla f(x_k)^T d + \tfrac{1}{2} d^T M_k d \\ \text{subject to} \quad & c(x_k) + A_k^T d = 0 \\ & \|d\|_2 \le \Delta_k \end{aligned} \tag{1.2}$$

where M_k is the Hessian of the Lagrangian function $L(x, \lambda) = f(x) + \lambda^T c(x)$ at (x_k, λ_k) or an approximation to it, $A_k = \nabla c(x_k)$, and $\Delta_k > 0$. However, a difficulty is that problem (1.2) may be infeasible.

To overcome this difficulty, in [4] and [28] the authors replace the equality constraint in (1.2) by a parameterized constraint, $\alpha c(x_k) + A_k^T d = 0$, where $0 < \alpha \le 1$. The resultant subproblem is always feasible if α is chosen properly. However, it is not clear how to choose the parameter α effectively in practice. In [6] the equality constraint in (1.2) is relaxed differently. The authors of [6] substitute an inequality constraint $\|c(x_k) + A_k^T d\|_2 \le \theta_k$ for the equality constraint in (1.2), where $\theta_k > 0$ tends to zero as $k \to \infty$. A similar method is presented in [26]. But these approaches are mainly of theoretical interest because no efficient procedure has been designed for solving the subproblems derived from the above relaxations. More recently, an algorithm aiming at the infeasibility problem was introduced in [21].

A rather different trust-region algorithm was proposed in [24] where a trust-region step is decomposed into two parts—a "normal" step and a "tangential" step (here normal and tangential are with respect to the constraint surface)—which are computed by solving two trust-region subproblems at each iteration. A parameter $r \in (0, 1)$ associated with the trust-region radius is determined to give the proportion of the sizes of the normal step and tangential step.

Our algorithm solves (1.1) by minimizing a sequence of quadratic penalty functions $\{p_{\mu_i}(x) = f(x) + \frac{1}{2\mu_i}\|c(x)\|_2^2\}$ where $\{\mu_i > 0\}$ monotonically decreases to zero. To obtain possible fast local convergence, we use a specially designed trust-region method to minimize $p_{\mu_i}(x)$ for fixed $\mu_i > 0$. The trust-region subproblems have no equality constraints hence the infeasibility difficulty will not occur. At each iteration, we introduce a local change of variables that forces the iterates to approximately follow the constraint level set. A key feature of this strategy is that the value of $\|c(x)\|_2$ changes only slightly along the level set and therefore long steps may be taken.

Hempel [20] presents an algorithm that uses this idea to formulate two coupled trust-region subproblems and to update the current iterate by solving the two subproblems. Under some strict assumptions, the algorithm in [20] is globally convergent with a local R-superlinear convergence rate. A related algorithm is proposed in [9].

The change of variables also leads to an efficient block diagonal approximation to the Hessian matrix of the quadratic penalty function. As a consequence, our algorithm obtains

the next iterate at each iteration in two parts: a Newton step with a line search is calculated in the subspace normal to the constraints and a trust-region subproblem is (approximately) solved in the subspace tangent to the constraints. Due to the line search along the normal direction, the algorithm automatically determines the sizes of the normal step and the tangential step without introducing any new proportion parameters. We solve the trust-region subproblem approximately using a two-dimensional subspace approach [5]. Improvement is measured using the quadratic penalty function $p_{\mu_i}(x)$: From one iteration to the next the algorithm always reduces the value of the quadratic penalty function for a fixed μ_i. The decrease of $p_{\mu_i}(x)$ leads to the global convergence of the algorithm.

The paper can be outlined as follows: In Section 2 we introduce the local change of variables and describe our algorithm. Under appropriate assumptions, we show in Section 3 that the sequence generated by our algorithm converges to a point satisfying the second-order necessary optimality conditions. In Section 4 some preliminary numerical results are illustrated and discussed. Finally we make some concluding remarks in Section 5.

We use $\|\cdot\|$ to denote $\|\cdot\|_2$ except otherwise specified.

2. Algorithm

As mentioned in § 1, our strategy for solving problem (1.1) is to minimize a sequence of penalty functions $\{p_{\mu_i}(x)\}$ as μ_i tends to zero. Under certain assumptions the limit point of the sequence of minimizers of $p_{\mu_i}(x)$ is a local minimizer of (1.1). Unfortunately, unconstrained minimization techniques often exhibit slow convergence when applied to the quadratic penalty function. One reason for the slow convergence is that the Hessian of $p_{\mu_i}(x)$ is dominated, in some directions, by the constraint gradients when both μ_i and $\|c(x)\|$ are small and the iterate x is far away from a local minimizer of (1.1). This causes most unconstrained minimization methods to compute steps almost entirely in the null space of $\nabla c(x)$, and to take very small steps to ensure that $p_{\mu_i}(x)$ decreases.

In this section, we develop a specially designed trust-region method for minimizing $p_{\mu_i}(x)$. At each iteration a local change of variables is used to force the correction step to move along a curve which approximately follows the constraint level set. This technique provides a remedy for the slow convergence problem.

We introduce the change of variables as follows. At the current point $x_k \in \Re^n$, let $A_k = \nabla c(x_k) \in \Re^{n\times m}$. Suppose that the QR factorization of A_k is

$$A_k = Q_k \bar{R}_k = [Y_k Z_k] \begin{bmatrix} R_k \\ 0 \end{bmatrix} = Y_k R_k$$

where $Q_k \in \Re^{n\times n}$ is orthogonal and R_k is an $m \times m$ upper triangular matrix. It is assumed that A_k has full column rank so that R_k is nonsingular. In this case, $Y_k \in \Re^{m\times n}$ and $Z_k \in \Re^{(n-m)\times n}$. Let

$$u_k(h) = x_k + s_k(h) \tag{2.3}$$

for $h \in \Re^{n-m}$ where

$$s_k(h) = Z_k h + Y_k R_k^{-T}[c(x_k) - c(x_k + Z_k h)]. \tag{2.4}$$

The following result illustrates important properties of the penalty function $p_\mu(x)$ and the constraint function $c(x)$ along the path $u_k(h)$.

Lemma 2.1. *Let* $p_\mu(x) = f(x) + \frac{1}{2\mu}\|c(x)\|_2^2$ *be the quadratic penalty function and let* $u_k(h)$ *be defined in* (2.3). *Then*

$$\begin{cases} \nabla_h p_\mu(u_k(0)) = Z_k^T \nabla f(x_k) \\ \nabla_h^2 p_\mu(u_k(0)) = Z_k^T \nabla^2 L(x_k) Z_k \equiv H_k \end{cases}$$

where $\nabla^2 L(x_k) = \nabla^2 f(x_k) + \sum_{j=1}^m (\lambda_k)_j \nabla^2 c_j(x_k)$ *and* $(\lambda_k)_j$ *is the jth component of the (least-squares) Lagrangian multiplier* $\lambda_k = -R_k^{-1} Y_k^T \nabla f(x_k)$. *Moreover, if the second-order derivative of c is Lipschitz continuous, then*

$$c(u_k(h)) = c(x_k) + \mathcal{O}(\|h\|^3) \quad \text{as } h \to 0.$$

Proof: See Lemma 2.1 in [9] □

Relation (2.5) tells us that the reduced gradient and the reduced Hessian of the Lagrangian function (with respect to x) are equal to the gradient and the Hessian of the penalty function (with respect to h), respectively. This property will be used in our algorithm.

Equality (2.6) indicates that along the path $u_k(h)$ the value of $\|c(x)\|$ varies only slightly around $h = 0$. It motivates us to follow the path when updating the iterates. Note that the Lipschitz continuity of the second-order derivatives of c is not required in our algorithm or in the convergence analysis.

In order to reduce $\|c(x)\|$ in an appropriate manner, it may also be necessary to move along the normal direction $Y_k v$ for some $v \in \Re^m$. Therefore, we consider the function $\hat{p}_\mu(h, v) = p_\mu(u_k(h) + Y_k v)$ for $h \in \Re^{n-m}$ and $v \in \Re^m$. It follows from the chain rule that

$$\nabla_{h,v} \hat{p}_\mu(0,0) = \begin{bmatrix} \nabla_h \hat{p}_\mu(0,0) \\ \nabla_v \hat{p}_\mu(0,0) \end{bmatrix} \quad \begin{bmatrix} Z_k^T \nabla p_\mu(x_k) \\ Y_k^T \nabla p_\mu(x_k) \end{bmatrix}$$

and

$$\nabla_{h,v}^2 \hat{p}_\mu(0,0) = \begin{bmatrix} Z_k^T \nabla^2 L(x_k) Z_k & Z_k^T W_k Y_k \\ Y_k^T W_k Z_k & Y_k^T W_k Y_k + \frac{1}{\mu} R_k R_k^T \end{bmatrix}$$

where $W_k = \nabla^2 f(x_k) + \sum_{j=1}^m \frac{c_j(x_k)}{\mu} \nabla^2 c_j(x_k)$.

The standard trust-region subproblem associated with $\hat{p}_\mu$ at $(h, v) = (0, 0)$ is

$$\begin{aligned} &\text{minimize} && \nabla_{h,v} \hat{p}_\mu(0,0)^T \hat{s} + \frac{1}{2} \hat{s}^T \nabla_{h,v}^2 \hat{p}_\mu(0,0)\, \hat{s} \\ &\text{subject to} && \|\hat{s}\|_2 \le \Delta \end{aligned}$$

where $\hat{s} = (h^T, v^T)^T$. It is well-known (e.g., [16]) that problem (.7) can be characterized by a linear system of the form

$$\begin{bmatrix} Z_k^T \nabla^2 L(x_k) Z_k + \alpha I_n & Z_k^T W_k Y_k \\ Y_k^T W_k Z_k & W_k Z_k + \frac{1}{\mu} R_k R_k^T + \alpha I, \end{bmatrix} \begin{bmatrix} h \\ \end{bmatrix} = \begin{bmatrix} Z_k^T \nabla p_\mu(x_k) \\ Y_k^T \nabla p_\mu(x_k) \end{bmatrix}$$

where $\alpha \geq 0$ and the coefficient matrix $(\nabla^2_{h,v} p_\mu(0,0) + \alpha I_n)$ of system (2.8) is positive semi-definite.

By Taylor's theorem, $Z_k^T \nabla p_\mu(x_k) + Z_k^T W_k Y_k v \approx Z_k^T \nabla p_\mu(x_k + Y_k v) \approx Z_k^T \nabla f(x_k + Y_k v)$. Thus, the upper part of system (2.8) can be approximately written as $(Z_k^T \nabla^2 L(x_k) Z_k + \alpha I_{n-m}) h = -Z_k^T \nabla f(x_k + Y_k v)$.

In addition, since $A_k = Y_k R_k$ and $\lambda_k = -R_k^{-1} Y_k^T \nabla f(x_k)$, we have

$$Y_k^T \nabla p_{\ } \ x_k) \quad R_k R_k^{-1} Y \left[\nabla \ x_k) + A_k \frac{c(x_k)}{\mu} \right] \quad {}_{\mu} R_k[\mu \lambda_k \quad c(x_k)]$$

As $\mu \to 0$, the matrix $\frac{1}{\mu} R_k R_k^T$ plays a dominant role in the lower part of system (2.8). Thus, system $\frac{1}{\mu} R_k R_k^T v = -Y_k^T \nabla p_\mu(x_k) = \frac{1}{\mu} R_k[\mu \lambda_k - c(x_k)] \approx -\frac{1}{\mu} R_k c(x_k)$ is an approximation to the lower part of system (2.8).

Therefore, system (2.8), hence problem (2.7), can be approximated by

$$\begin{bmatrix} B_k + \alpha I & 0 \\ 0 & R_k^T \end{bmatrix} \begin{bmatrix} h \\ v \end{bmatrix} = \begin{bmatrix} Z_k^T \nabla f(x_k + Y_k v \\ c(x_k) \end{bmatrix}$$

where B_k is the reduced Hessian $H_k = Z_k^T \nabla^2 L(x_k) Z_k$ of the Lagrangian function or an approximation to it. Note that the penalty parameter μ is absent from system (2.10).

System (2.10) suggests that trust-region subproblem (2.7) can be (approximately) solved in two steps. First solve

$$R_k^T v \quad \cdot c(x_k$$

for v, determine a step lenght β by a line search procedure as described below, then solve

$$\begin{aligned} &\text{minimize} \quad \left(Z_k^T \nabla f(x_k + \beta Y_k v)\right)^T h + \frac{1}{2} h^T B_k h \\ &\text{subject to} \quad \|h\|_2 \leq \Delta_k \end{aligned}$$

for some $\Delta_k > 0$, and finally set $x_{k+1} = u_k(h) + \beta Y_k v$, where h solves (2.12) and v is given by (2.11).

To ensure that the penalty function $p_\mu(x)$ decreases, we perform the following back-tracking line search to choose the step length β along the direction $Y_k v$ at x_k.

Set β

2. Until the line search condition

$$p_\mu(x_k + \beta Y_k v) - p_\mu(x_k) \le \sigma\beta \nabla p_\mu(x_k)^T Y_k v \tag{2.13}$$

is satisfied, choose a new $\beta \in [\tau_1\beta, \tau_2\beta]$,

where $0 < \sigma < 1$ and $0 < \tau_1 < \tau_2 < 1$ are given. As we will see in Lemmas 3.1 and 3.2, the line search will always terminate and guarantee that the penalty function $p_\mu(x)$ decreases.

An important feature of our algorithm is that, for fixed μ, the normal step v is calculated only when x_k is outside an envelope around the surface $c(x) = 0$. More precisely, v is computed only when $\|c(x_k)\| > \mu \cdot \max\{\|\lambda_k\|/\sigma, 1\}$. Otherwise, the subproblem (2.12) is solved for h with $v = 0$, and the next iterate will be $x_{k+1} = u_k(h) = x_k + s_k(h)$. Due to (2.6), $\|c(x)\|$ varies only mildly along the curve $u_k(h)$ beginning at $h = 0$.

At the beginning of this section we mentioned that the columns of the matrix Z_k form an orthonormal basis for the null space of A_k^T. However, this does not completely specify Z_k. Moreover, if the choice of the basis for the null space changes significantly from one iterate to the next, convergence of our algorithm cannot be achieved. Coleman and Sorensen [10] investigate this issue and suggest several procedures for computing Z_k when performing a *QR* factorization of A_k. Those procedures guarantee that $Z(x)$ varies smoothly, locally. In our algorithm, we use the approach of Coleman and Sorensen to compute $Z_k = Z(x_k)$. For details, see [10].

Our algorithm, Algorithm 2.1, is described in figure 1. For simplicity of discussion, the reduced Hessian H_k will be used for the rest of the paper instead of an approximation B_k. Since the algorithm includes outer iterations and inner iterations, we use $x_i^{(l)}$ instead of x_k to denote the iterates, where i is the outer loop index (corresponding to μ_i) and l is the inner loop index (for fixed μ_i). The intermediate iterates $x_i^{(l)} + \beta Y_i^{(l)} v$ are indexed by $x_i^{(l+)}$. Accordingly, we may have $Z_i^{(l)}$, $Z_i^{(l+)}$, $Y_i^{(l)}$, $Y_i^{(l+)}$, $H_i^{(l)}$, and $H_i^{(l+)}$ etc.

For a fixed parameter μ_i an inner loop is employed to approximately minimize $p_{\mu_i}(x)$ until the following conditions are satisfied.

$$\begin{cases} a) & \left\|(Z_i^{(l)})^T \nabla f(x_i^{(l)})\right\| \le \mu_i^{1/2} \\ b) & \left\|c(x_i^{(l)})\right\| \le \Lambda_i^{(l)} \mu_i, \\ c) & H_i^{(l)} + \theta_i I \ge 0, \end{cases} \tag{2.14}$$

where $\Lambda_i^{(l)} = \max\{\|\lambda_i^{(l)}\|/\sigma, 1\}$ and $\theta_i > 0$ tends to zero as $i \to \infty$. After (2.14) is satisfied, the parameter μ_i is reduced to μ_{i+1} and a new inner loop starts. This process continues as the parameter μ_i goes to zero.

In Algorithm 2.1, $q_i^{(l)}(h) = g(x_i^{(l+)})^T h + \frac{1}{2} h^T H_i^{(l+)} h$ where

$$g(x) := Z(x)^T \nabla f(x),$$

and

$$x_i^{(l+)} = x_i^{(l)} \quad \text{if (2.14)-b) holds,} \quad \text{or } x_i^{(l+)} = x_i^{(l)} + \beta Y_i^{(l)} v \quad \text{otherwise}$$

Let $\mu_{\min} > 0$ be the tolerance for μ_i. Let $\kappa > 0$, $0 < \rho < 1$, $0 < \sigma < 1$, $0 < \tau_1 < \tau_2 < 1$, and $0 < \eta_1 < \eta_2 < 1$. Choose $\mu_1 > \mu_{\min}$, $0 < \theta_0 < 1$, an initial point $x_0^* \in \Re^n$, and an initial trust region radius $\Delta_0^* > 0$. Set $i = 1$.

```
Do
    Set l := 0; x_i^(0) := x_{i-1}^*; and Δ_i^(0) = min{2Δ_{i-1}^*, κμ_i^{2/5}};
    while any of the criteria in (2.14) does not hold (at x_i^(l))
        if (2.14)-b) does not hold
            Solve (R_i^(l))^T v_i^(l) = -c(x_i^(l));
            Find the step length β_i^(l) > 0 by (2.13);
            x_i^(l+) := x_i^(l) + β_i^(l) Y_i^(l) v_i^(l);
        else
            x_i^(l+) := x_i^(l);    (i.e., v_i^(l) = 0)
        end;
        Compute h_i^(l) satisfying the three conditions (2.15) - (2.17);
        Calculate the ratio r_i^(l) = [p_{μ_i}(u_i^(l+)(h_i^(l))) - p_{μ_i}(u_i^(l+)(0))] / q_i^(l)(h_i^(l));
        while r_i^(l) < η_1
            Set Δ_i^(l) := Δ_i^(l)/4;
            Solve (2.12) to find an h_i^(l) satisfying the conditions (2.15) - (2.17)
            Calculate the ratio r_i^(l);
        end;
        if r_i^(l) > η_2 ,
            Δ_i^(l) := min{2Δ_i^(l), κμ_{i-1}^{2/5}};
        end;
        Δ_i^(l+1) = Δ_i^(l);
        x_i^(l+1) = u_i^(l+)(h_i^(l));
        l := l + 1;
    end;
    x_i^* := x_i^(l);  Δ_i^* := Δ_i^(l);  μ_{i+1} := min{ρμ_i, μ_i^{6/5}};
    θ_{i+1} := min{ρθ_i, θ_i^{6/5}};  i := i + 1;
while    μ_{i-1} > μ_min;
Set  x* := x_i^* and STOP;
```

Figure 1. Algorithm 2.1

In each inner loop we solve (2.11) when (and only when) (2.14)-b) does not hold. Next we approximately solve the trust-region subproblem $\min\{q_i^{(l)}(h) : \|h\|_2 \le \Delta_i^{(l)}\}$ by computing an $h = h_i^{(l)}$ with $\|h\| \le \Delta_i^{(l)}$ satisfying the following three conditions (2.15)–(2.17).

Condition 1. There is a $\gamma_1 > 0$ such that

$$q_i^{(l)}(h) \le -\gamma_1 \left\| g\left(x_i^{(l+)}\right) \right\| \min\left\{ \Delta_i^{(l)}, \left\| g\left(x_i^{(l+)}\right) \right\| / \left\| H_i^{(l+)} \right\| \right\}. \tag{2.15}$$

Condition 2. There is a $\gamma_2 > 0$ such that

$$q_i^{(l)}(h) \le \left(\Delta_i^{(l)}\right)^2 \nu_{\min}\left(H_i^{(l+)}\right) / \gamma_2 \tag{2.16}$$

where $\nu_{\min}(H_i^{(l+)})$ is the smallest eigenvalue of the matrix $H_i^{(l+)}$

Condition 3. When $H_i^{(l+)}$ is positive definite and $\|(H_i^{(l+)})^{-1}\bar{g}\| \leq \Delta_i^{(l)}$ for all $\bar{g} \in \Re^{n-m}$ satisfying $\|g(x_i^{(l+)}) - \bar{g}\| \leq \epsilon \|g(x_i^{(l+)})\|$,

$$\left\|H_i^{(l+)}h + g\left(x_i^{(l+)}\right)\right\| \leq \epsilon \left\|g\left(x_i^{(l+)}\right)\right\|$$

where $0 \leq \epsilon < 1/5$ is given.

Note that, if $\epsilon = 0$ in (2.17), h is the exact solution to the linear system $H_i^{(l+)}h = -g(x_i^{(l+)})$. As showed by Schultz et al. in [27], when $\epsilon = 0$, any h satisfying Conditions 1, 2 and 3 approximately solves the trust-region problem. In addition, the exact solution to the trust-region subproblem satisfies all these conditions.

It follows from Algorithm 2.1 that

$$\left\|h_i^{(l)}\right\| \leq \Delta_i^{(l)} \leq \kappa \mu_{i-1}^{2/5} \tag{2.18}$$

and

$$\mu_i = \mu_{i-1}^{6/5} \quad \text{for all } i \text{ sufficiently large} \quad \text{and } \mu_i \leq \rho\, \mu_{i-} \quad \text{for every } i \tag{2.19}$$

The rule to update μ_i follows from the analysis in [19] where it is required that $\mu_i \to 0$ at least linearly but slower than quadratic. Our numerical experiments suggest that μ_i should not approach zero too quickly. In theory, the sequence $\{\theta_i\}$ can be any positive sequence tending to zero. The choice in Algorithm 2.1 is the one we used in our numerical experiment.

3. Global convergence

In this section we analyze the global convergence properties of Algorithm 2.1. We begin this section with our assumptions. Under these assumptions we show that all limit points of the sequence generated by Algorithm 2.1 satisfy the first order necessary optimality conditions to (1.1). Then we prove that if the number of limit points is finite, the primary sequence converges to a point satisfying the second-order necessary optimality conditions.

Assumption 3.1. The sequence $\{x_i^{(l)}\}$ generated by Algorithm 2.1 is contained in an open convex set D and the level set $\{x \mid p_{\mu_0}(x) \leq p_{\mu_0}(x_0)\}$ is bounded. In addition, the following properties hold.

1. The functions $f : \Re^n \to \Re$, and $c : \Re^n \to \Re^m$ and their first and second derivatives are uniformly bounded in norm over D.
2. The matrix $A(x)$ has full column rank, and there is a constant K_0 such that

$$\|A(x)[A(x)^T A(x)]^{-1}\| \leq K_0$$

for all $x \in D$.

Note that (3.20) implies $\|Y_i^{(l)}(R_i^{(l)})^{-T}\| \leq K_0$ for all i and l.

Recall that the vector $v_i^{(l)}$ is calculated only when (2.14)-b) does not hold. The next two results show that $Y_i^{(l)}v_i^{(l)}$ is a descent direction for p_{μ_i} at $x_i^{(l)}$, the line search (2.13) will terminate, and the sequence $\{p_{\mu_i}(x_i^{(l)})\}$ is decreasing for fixed i.

Lemma 3.1. *Suppose in Algorithm* 2. $\|c(x_i^{(l)})\| > \mu_i \max\{\|\lambda_i^{(l)}\|/\sigma, 1\}$. *Then*

$$\nabla p_{\mu_i}(x_i^{(l)})^T Y_i^{(l)} v_i^{(l)} = c(x_i^{(l)})^T \lambda_i^{(l)} - \frac{1}{\mu_i}\|c(x_i^{(l)})\|^2 < 0.$$

Consequently the line search condition (2.13) *will be satisfied.*

Proof: The equality in (3.21) follows from (2.9) and (2.11). The inequality holds because $0 < \sigma < 1$ and the condition $\|c(x_i^{(l)})\| > \mu_i \max\{\|\lambda_i^{(l)}\|/\sigma, 1\}$ implies

$$c(x_i^{(l)})^T \lambda_i^{(l)} - \frac{1}{\mu_i}\|c(x_i^{(l)})\|^2 < c(x_i^{(l)})^T \lambda_i^{(l)} - \frac{1}{\sigma}\|c(x_i^{(l)})\|\|\lambda_i^{(l)}\|$$

$$\leq \left(1 - \frac{1}{\sigma}\right)\|c(x_i^{(l)})\|\|\lambda_i^{(l)}\| \tag{3.23}$$

$$< 0. \tag{3.24}$$

Now from Theorem 6.3.2 in [13], the line search condition (2.13) will be satisfied by some $\beta_i^{(l)}$. □

Lemma 3.2. *The sequence* $\{x_i^{(l)}$ *generated by Algorithm* 2.1 *satisfies*

$$p_{\mu_i}(x_i^{(l+1)}) - p_{\mu_i}(x_i^{(l)}) \leq 0.$$

Proof: Since $x_i^{(l+1)} = u_i^{(l+)}(h_i^{(l)})$ and $x_i^{(l+)} = u_i^{(l+)}(0)$, it follows from the definition of $r_i^{(l)}$ and (2.15) that

$$p_{\mu_i}(x_i^{(l+1)}) - p_{\mu_i}(x_i^{(l+)}) = r_i^{(l)} q_i^{(l)}(h_i^{(l)}) \leq 0.$$

If (2.14)-b) holds, then $x_i^{(l+)} = x_i^{(l)}$ and inequality (3.26) yields (3.25). If (2.14)-b) does not hold, we have from (2.13) and (3.21) that

$$p_{\mu_i}(x_i^{(l+)}) - p_{\mu_i}(x_i^{(l)}) \leq \sigma\beta_i^{(l)}\left[c(x_i^{(l)})^T \lambda_i^{(l)} - \frac{1}{\mu_i}\|c(x_i^{(l)})\|^2\right] < 0 \tag{3.27}$$

Combining inequalities (3.26) and (3.27), we obtain (3.25). □

In Algorithm 2.1, for any fixed μ_i the penalty function $p_{\mu_i}(x)$ is minimized until (2.14) is satisfied. In Lemmas 3.4 and 3.5 we illustrate that for any fixed μ_i, (2.14) holds after a

finite number of inner iterations. Before doing that, we cite a lemma from [9] which says that the step lengths $\{\beta_i^{(l)}\}$ are uniformly bounded away from zero.

Lemma 3.3. *Under Assumption* 3 *there is a constant* $\beta > 0$, *such that for all* i *and* l *the step length* $\beta_i^{(l)}$ *satisfies*

$$\beta_i^{(l)} \geq \beta.$$

Proof: See Lemma 3 in [9]. □

From (3.25) and (3.27), for any $\bar{l} > l$,

$$p_{\mu_i}(x_i^{(\bar{l})}) - p_{\mu_i}(x_i^{(l)}) \leq p_{\mu_i}(x_i^{(l+)}) - p_{\mu_i}(x_i^{(l)}) \leq -\sigma(1-\sigma)\beta_i^{(l)} \frac{\|c(x_i^{(l)})\|^2}{\mu_i} \tag{3.29}$$

Thus, for fixed μ_i, inequality (2.14)-b) holds after a finite number of inner iterations. Otherwise, $\|c(x_i^{(l)})\| > \Lambda_i^{(l)}\mu_i$, and we would have from (3.29) that

$$p_{\mu_i}(x_i^{(\bar{l})}) - p_{\mu_i}(x_i^{(l)}) \leq -\sigma(1-\sigma)\beta_i^{(l)}(\Lambda_i^{(l)})^2\mu_i \leq -\sigma(1-\sigma)\beta\mu_i < 0. \tag{3.30}$$

Inequality (3.30) contradicts the fact that $p_{\mu_i}(x)$ is bounded below for fixed μ_i. Therefore we obtain the following result.

Lemma 3.4. *Suppose Assumption* 3.1 *is satisfied. Suppose* $\{x_i^{(l)}\}$ *is generated by Algorithm* 2.1. *Then for any fixed* i *there exists an integer* $\bar{l}_i > 0$, *such that for all* $l \geq \bar{l}_i$

$$\|c(x_i^{(l)})\| \leq \Lambda_i^{(l)}\mu_i. \tag{3.31}$$

Next we show that (2.14)-a) and (2.14)-c) will be satisfied. To this end, let $\Delta > 0$ be a trust-region radius. Suppose that h satisfies Conditions 1–3 in Section 2 with $\Delta_i^{(l)} = \Delta$. Using Taylor's theorem, we obtain that

$$\begin{aligned} &p_{\mu_i}(u_i^{(l+)}(h)) - p_{\mu_i}(u_i^{(l+)}(0)) \\ &\quad = \nabla_h p_{\mu_i}(u_i^{(l+)}(0))^T h + h^T \left[\int_0^1 \nabla_h^2 p_{\mu_i}(u_i^{(l+)}(\tau h))(1-\tau)\,d\tau\right] h \\ &\quad = q_i^{(l)}(h) + h^T \left\{\int_0^1 [\nabla_h^2 p_{\mu_i}(u_i^{(l+)}(\tau h)) - \nabla_h^2 p_{\mu_i}(u_i^{(l+)}(0))](1-\tau)\,d\tau\right\} h \end{aligned}$$

Subtracting $q_i^{(l)}(h)$ from both sides and then dividing both sides by $q_i^{(l)}(h)$, we have

$$\begin{aligned} |r_i^{(l)} - 1| &= |[p_{\mu_i}(u_i^{(l+)}(h)) - p_{\mu_i}(u_i^{(l+)}(0))] - q_i^{(l)}(h)| / |q_i^{(l)}(h)| \\ &\leq \frac{\|h\|^2}{|q_i^{(l)}(h)|} \left\| \int_0^1 [\nabla_h^2 p_{\mu_i}(u_i^{(l+)}(\tau h)) - \nabla_h^2 p_{\mu_i}(u_i^{(l+)}(0))](1-\tau)\,d\tau \right\| \end{aligned}$$

Inequality (3.32) is used in the next lemma.

Lemma 3.5. *Suppose Assumption* 3.1 *is satisfied. Suppose* $\{x_i^{(l)}\}$ *is generated by Algorithm* 2.1. *Then for fixed* $\mu_i > 0$, *there exists a subsequence of* $\{x_i^{(l)}\}$, $l \in \mathcal{L}$, *such that*

$$\lim_{\substack{l\in\mathcal{L}\\ l\to\infty}} \|g(x_i^{(l)})\| = 0$$

and

$$\lim_{\substack{l\in\mathcal{L}\\ l\to\infty}} \nu_{\min}(H(x_i^{(l)})) \geq 0,$$

where $g(x) = Z(x)^T \nabla f(x)$, $H(x) = Z(x)^T \nabla^2 L(x) Z(x)$ *and* $\nu_{\min}(H(x_i^{(l)}))$ *denotes the smallest eigenvalue of* $H(x_i^{(l)})$.

Proof: Due to Lemma 3.4, we can assume that $x_i^{(l+)} = x_i^{(l)}$ without loss of generality. There are two possible cases: either

1. $\inf_l \Delta_i^{(l)} = \bar{\Delta} > 0$, or
2. There exists an integer set $\bar{\mathcal{L}}$ such that $\Delta_i^{(l+1)} < \Delta_i^{(l)}$ and $\Delta_i^{(l)} \to 0$ for $l \in \bar{\mathcal{L}}$

In case 1, suppose there exists an $\epsilon > 0$ such that, for all l, $\|g(x_i^{(l)})\| \geq \epsilon$. Then, because $\|H_i^{(l)}\| \leq K$, it follows from (2.15) in Condition 1 that

$$q_i^{(l)}(h_i^{(l)}) \leq -\gamma_1 \epsilon \min\{\bar{\Delta}, \epsilon/K\} \equiv -\bar{\epsilon}$$

Recall that according to Algorithm 2.1, we have that for all l

$$p_{\mu_i}(x_i^{(l+1)}) - p_{\mu_i}(x_i^{(l)}) \leq r_i^{(l)} q_i^{(l)}(h_i^{(l)}) \leq \eta_1 q_i^{(l)}(h_i^{(l)}) \leq -\eta_1 \bar{\epsilon} < 0$$

But by the definition of $p_{\mu_i}(x)$ we know that $p_{\mu_i}(x)$ is bounded below, thus

$$p_{\mu_i}(x_i^{(l+1)}) - p_{\mu_i}(x_i^{(l)}) \to 0 \quad \text{as } l \to \infty.$$

This contradiction establishes that there exists an integer set $\tilde{\mathcal{L}}$ such that

$$\lim_{l\to\infty} \{g(x_i^{(l)})\}_{l\in\tilde{\mathcal{L}}} = 0.$$

Now, suppose there exists an $\epsilon > 0$ such that, for all $l \in \tilde{\mathcal{L}}$, $\nu_{\min}(H(x_i^{(l)})) \leq -\epsilon < 0$. Then, (2.16) in Condition 2 yields $q_i^{(l)}(h_i^{(l)}) \leq -\frac{(\Delta_i^{(l)})^2}{\gamma_2}\epsilon \leq -\frac{\bar{\Delta}^2}{\gamma_2}\epsilon < 0$. Similar to (3.35), this inequality contradicts the fact that $p_{\mu_i(x)}$ is bounded below. Therefore (3.33) and (3.34) hold with an integer set $\mathcal{L} \subset \tilde{\mathcal{L}}$.

In case 2, by Assumption 3.1, $\{x_i^{(l)}\}_{l\in\bar{\mathcal{L}}}$ is in a compact set D. Thus there exists a convergent subsequence. Without loss of generality, we assume that

$$\Delta_i^{(l)} \to 0 \quad \text{and} \quad x_i^{(l)} \to \bar{x} \quad \text{for } l \in \bar{\mathcal{L}}.$$

Recall that in Algorithm 2. if the radius $\Delta_i^{(l)}$ decreases, there must be a previously tried radius $\bar{\Delta}_i^{(l)}$ for which

$$\Delta_i^{(l)} = \frac{1}{4}\bar{\Delta}_i^{(l)} \quad \text{and} \quad r_i^{(l)} = r_i^{(l)}\big(\bar{\Delta}_i^{(l)}\big) < \eta_1$$

Since $\Delta_i^{(l)} \to 0$, we have

$$r_i^{(l)}\big(\bar{\Delta}_i^{(l)}\big) < \eta_1 \quad \text{and} \quad \bar{\Delta}_i^{(l)} \to 0 \quad \text{for } l \in \bar{\mathcal{L}}. \tag{3.36}$$

Suppose $\|g(\bar{x})\| = 2\epsilon > 0$, then for $l \in \bar{\mathcal{L}}$ sufficiently large, $\|g(x_i^{(l)})\| > \epsilon$. It follows from (2.15) that for $l \in \bar{\mathcal{L}}$

$$q_i^{(l)}\big(h_i^{(l)}\big) \le -\gamma_1\epsilon \min\{\bar{\Delta}_i^{(l)}, \epsilon/K\} \le -\gamma_1\epsilon\bar{\Delta}_i^{(l)}$$

since $\bar{\Delta}_i^{(l)} \to 0$. Thus, from (3.32), for $l \in \bar{\mathcal{L}}$,

$$\begin{aligned}\big|r_i^{(l)}\big(\bar{\Delta}_i^{(l)}\big) - 1\big| \le \frac{\bar{\Delta}_i^{(l)}}{\gamma_1\epsilon}\int_0^1 \big\|&\nabla_h^2 p_{\mu_i}\big(u_i^{(l+)}(\tau h_i^{(l)})\big) \\ &-\nabla_h^2 p_{\mu_i}\big(u_k^{(l+)}(0)\big)\big\|(1-\tau)\,d\tau \to 0\end{aligned}$$

This contradicts (3.36). Therefore (3.33) holds with $\mathcal{L} = \bar{\mathcal{L}}$.

To show (3.34), suppose $\nu_{\min}(H(\bar{x})) \le -2\epsilon < 0$. Then for $l \in \bar{\mathcal{L}}$ sufficiently large, $\nu_{\min}(H(x_i^{(l)})) \le \epsilon$. Thus, (2.16) implies that for $l \in \bar{\mathcal{L}}$

$$q_i^{(l)}\big(h_i^{(l)}\big) \le -\frac{\big(\bar{\Delta}_i^{(l)}\big)^2}{\gamma_2}\epsilon$$

It follows from (3.32) that for $l \in \bar{\mathcal{L}}$

$$\begin{aligned}\big|r_i^{(l)}\big(\bar{\Delta}_i^{(l)}\big) - 1\big| \le \frac{\gamma_2}{\epsilon}\int_0^1 \big\|&\nabla_h^2 p_{\mu_i}\big(u_i^{(l+)}(\tau h_i^{(l)})\big) \\ &-\nabla_h^2 p_{\mu_i}\big(u_i^{(l+)}(0)\big)\big\|(1-\tau)\,d\tau \to 0\end{aligned}$$

since $\|h_i^{(l)}\| \le \|\bar{\Delta}_i^{(l)}\| \to 0$. This also contradicts (3.36). Therefore (3.34) holds with $\mathcal{L} = \bar{\mathcal{L}}$. □

Lemmas 3.4 and 3.5 indicate that for any fixed μ_i there exists an integer $\bar{l} > 0$ such that for all $l \ge \bar{l}$, the criteria in (2.14) are satisfied. From now on, for each fixed μ_i, we will

let l_i denote the first integer ≥ 0 for which (2.14) is satisfied. Therefore, Algorithm 2. generates a sequence

$$\{x_i^{(l)}\} = \{x_1^{(0)}, \quad x_1^{(l_1-1)}, x_2^{(0)}, \quad ., x_2^{(l_2-1)} \quad x_i^{(0)} \quad ., x_i^{(l_i-1)}$$

We reindex the sequence $\{x_i^{(l)}\}$ $(i = 1, 2, \ldots; 0 \leq l \leq l_i - 1)$ as $\{x_k\}$ $(k = 1, 2, \ldots)$. In other words, each given k has a one-to-one correspondence with a pair (i, l) such that $x_k = x_i^{(l)}$. It is clear that $i \to \infty$ as $k \to \infty$.

Lemma 3.6. *Suppose Assumption* 3 *is satisfied. Let* $\{x_k\} = \{x_i^{(l)}\}$ *be the sequence generated by Algorithm* 2.1. *Then*

$$\liminf_{k\to\infty}(\|Z(x_k)^T \nabla f(x_k)\| + \|c(x_k)\|) = 0.$$

$$\limsup_{k\to\infty} \nu_{\min}\big(Z(x_k)^T \nabla^2 L(x_k) Z(x_k) + \theta_i I\big) \geq 0.$$

Proof: Since (2.14) holds for $l = l_i$, μ_i tends to zero, $\{\Lambda_i^{(l_i)}\}$ is bounded (follows from (2.14) and the definition of $\Lambda_i^{(l)}$ after (2.14)), and θ_i tends to zero, we have

$$\lim_{i\to\infty} \big(\|Z(x_i^{(l_i)})^T \nabla f(x_i^{(l_i)})\| + \|c(x_i^{(l_i)})\|\big) = 0$$

$$\lim_{i\to\infty} \nu_{\min}\big(Z(x_i^{(l_i)})^T \nabla^2 L(x_i^{(l_i)}) Z(x_i^{(l_i)}) + \theta_i I\big) \geq 0$$

That completes the proof. □

From the proof of Lemma 3.5 we see that Condition 1 yields (3.33), and Condition 2 implies (3.34). Thus, if $h_i^{(l)}$ does not satisfy Conditions 2 and 3, we can still establish (3.33) from Condition 1. As we will see in the rest of this section, x_k converges to a point satisfying the first-order necessary optimality conditions if $h_i^{(l)}$ in Algorithm 2.1 satisfies Condition 1. Before we prove in Lemmas 3.8 and 3.9 that all limit points of $\{x_k\}$ satisfy the first-order necessary conditions, we cite another lemma from [9].

Lemma 3.7. *Suppose Assumption* 3.1 *is satisfied. Assume* $\{x_i^{(l)}$ *is generated by Algorithm* 2.1. *Then*

$$\sum_{i=1}^{\infty} \sum_{l=0}^{l_i-1} \big[p_{\mu_i}(x_i^{(l)}) - p_{\mu_i}(x_i^{(l+1)})\big] < +\infty. \tag{3.37}$$

Proof: See Lemma 3.4 in [9] □

Because of (3.37), we are able to prove in Lemmas 3.8 and 3.9 that any limit points of sequence $\{x_k\}$ described in Lemma 3.6 satisfy the first-order necessary optimality conditions.

Lemma 3.8. *Suppose Assumption* 3 *is satisfied. Let* $\{x_k\} = \{x_i^{(l)}\}$ *be the sequence generated by Algorithm* 2.1. *Then*

$$\lim_{k\to\infty} \|c(x_k)\| = 0. \tag{3.38}$$

Proof: Let

$$d_k = \begin{cases} \dfrac{\|c(x_k)\|^2}{\mu_i} - c(x_k)^T\lambda_k & \text{if } \|c(x_k)\| > \Lambda_k\mu_i; \\ 0 & \text{otherwise.} \end{cases}$$

By the definition of Λ_k (after (2.14)), $d_k \geq 0$. It is clear from Lemmas 3.1 and 3.4 that

$$\sigma\beta\, d_k \equiv \sigma\beta\, d_i^{(l)} \leq p_{\mu_i}(x_i^{(l)}) - p_{\mu_i}(x_i^{(l+)}) \leq p_{\mu_i}(x_i^{(l)}) - p_{\mu_i}(x_i^{(l+1)})$$

which, with (3.37), implies that

$$\sum_{k=1}^{\infty} d_k = \sum_{i=1}^{\infty}\sum_{l=0}^{l_i-1} d_i^{(l)} < +\infty \tag{3.39}$$

Therefore, $\lim_{k\to\infty} d_k = 0$.

Notice that since $\sigma\Lambda_k \geq \|\lambda_k\|$ and $\Lambda_k \geq 1$, we have that if $\|c(x_k)\| > \Lambda_k\mu_i$,

$$\begin{aligned} \|c(x_k)\| &= \frac{1}{(1-\sigma)\Lambda_k}[\Lambda_k\|c(x_k)\| - \sigma\Lambda_k\|c(x_k)\|] \\ &\leq \frac{1}{(1-\sigma)\Lambda_k}\left[\frac{\|c(x_k)\|^2}{\mu_i} - c(x_k)^T\lambda_k\right] \\ &\leq \frac{d_k}{(1-\sigma)\Lambda_k} \leq \frac{d_k}{1-\sigma} \end{aligned} \tag{3.40}$$

Therefore, $\|c(x_k)\| \leq \max\{\frac{d_k}{1-\sigma}, \Lambda_k\mu_i\}$, which implies (3.38) since $\mu_i \to 0$ and $d_k \to 0$. □

Lemma 3.9. *Suppose Assumption* 3 *is satisfied. Let* $\{x_k\} = \{x_i^{(l)}\}$ *be the sequence generated by Algorithm* 2.1. *Then*

$$\lim_{k\to\infty} \|Z(x_k)^T\nabla f(x_k)\| = 0. \tag{3.41}$$

Proof: Let $g(x) = Z(x)^T \nabla f(x)$ and $x_i^{(l_i+)}$ denote $x_{i+1}^{(0+)}$ (recall that $x_i^{(l_i)} = x_{i+1}^{(0)}$). We first prove by contradiction that $\|g(x_i^{(l+)})\| \to 0$. Without loss of generality, we assume that for any i there is an integer $k_i (0 \le k_i < l_i)$ such that $\|g(x_i^{(k_i+)})\| \ge \epsilon > 0$. Because of (3.20) and the definition of $v_i^{(l)}$, it follows from (3.38) and Taylor's theorem that for all $0 \le l \le l_i$ and all i,

$$\|g(x_i^{(l+)})\| = \|g(x_i^{(l)} + \beta_i^{(l)} Y_i^{(l)} v_i^{(l)})\| = \|g(x_i^{(l)})\| + \mathcal{O}(\|c(x_i^{(l)})\|)$$

Since $\|g(x_i^{(l_i)})\| \le \mu_i^{1/2}$ and $\mu_i \to 0$, (3.38) implies that $\|g(x_i^{(l_i+)})\| < \epsilon/2$ for i sufficiently large. Assume that $l = j_i$ is the nearest index to k_i in the set $\{l \in (k_i, l_i] : \|g(x_i^{(l)})\| < \epsilon/2\}$. Then $\|g(x_i^{(l)})\| \ge \epsilon/2$ for all l satisfying $k_i \le l < j_i$. Hence, from (2.15), for $k_i \le l \le j_i - 1$,

$$p_{\mu_i}(x_i^{(l)}) - p_{\mu_i}(x_i^{(l+1)}) \ge -r_i^{(l)} q_i^{(l)}(h_i^{(l)}) \ge -\eta_1 q_i^{(l)}(h_i^{(l)}) \ge \frac{\eta_1 \gamma_1 \epsilon}{2} \min\left\{\Delta_i^{(l)}, \frac{\epsilon}{2K}\right\} \tag{3.43}$$

Inequality (3.43) leads to

$$\begin{aligned} p_{\mu_i}(x_i^{(k_i)}) - p_{\mu_i}(x_i^{(j_i)}) &= \sum_{l=k_i}^{j_i-1} [p_{\mu_i}(x_i^{(l)}) - p_{\mu_i}(x_i^{(l+1)})] \\ &\ge \frac{\eta_1 \gamma_1 \epsilon}{2} \min\left\{\sum_{l=k_i}^{j_i-1} \Delta_i^{(l)}, \frac{\epsilon}{2K}\right\} \\ &\ge \frac{\eta_1 \gamma_1 \epsilon}{2} \min\left\{\sum_{l=k_i}^{j_i-1} \|h_i^{(l)}\|, \frac{\epsilon}{2K}\right\} \end{aligned}$$

Combining (3.37) and (3.44), we have that $\sum_{l=k_i}^{j_i-1} \|h_i^{(l)}\| \to 0$ as $i \to \infty$. Recall that $v_i^{(l)} = 0$ whenever (2.14)-b) holds. It follows from Assumption 3.1 (2), (2.11), (3.39), and (3.40) that

$$\sum_{l=k_i}^{j_i} \|v_i^{(l)}\| \le K_0 \sum_{l=k_i}^{j_i} \|c(x_i^{(l)})\| \le \frac{K_0}{-\sigma} \sum_{l=k_i}^{j_i} d_i^{(l)} \to 0, \quad \text{as } i \to \infty.$$

Therefore, the continuity of $u_k(h)$ implies that there exists a constant $K_1 > 0$ such that

$$\|x_i^{(k_i+)} - x_i^{(j_i+)}\| \le \sum_{l=k_i}^{j_i-1} \|x_i^{(l+)} - x_i^{(l+1+)}\| \le K_1 \left(\sum_{l=k_i}^{j_i-1} \|h_i^{(l)}\| + \sum_{l=k_i+1}^{j_i} \|v_i^{(l)}\| \right) \to 0$$

as $i \to \infty$. Due to the continuity of $Z(x)$ and $\nabla f(x)$ and (3.46), we have that

$$\left\| g\left(x_i^{(k_i+)}\right) - g\left(x_i^{(j_i+)}\right) \right\| \leq \epsilon/4$$

for all i sufficiently large. Therefore when i is sufficiently large,

$$\left\| g\left(x_i^{(k_i+)}\right) \right\| \leq \left\| g\left(x_i^{(k_i+)}\right) - g\left(x_i^{(j_i+)}\right) \right\| + \left\| g\left(x_i^{(j_i+)}\right) \right\| \leq \epsilon/4 + \epsilon/2 = 3\epsilon/4$$

This contradicts the assumption that $\|g(x_i^{(k_i+)})\| \geq \epsilon$. Therefore $\|g(x_i^{(l+)})\| \to 0$. Then (3.41) follows from (3.42) and (3.38). □

Now we further assume that the sequence $\{x_k\}$ has only a finite number of limit points. We end this section by showing in Theorem 3.1 that in this case there is actually only one limit point and the second-order necessary optimality conditions are satisfied at this point.

Theorem 3.1. *Let $\{x_k\} = \{x_i^{(l)}\}$ be the sequence generated by Algorithm* 2.1. *Suppose Assumption* 3.1 *is satisfied and there exist only a finite number of limit points to $\{x_k\}$. Then*

$$\lim_{k \to \infty} x_k = x_*$$

where x_* is a point at which the second-order necessary optimality conditions are satisfied.

Proof: We first prove by contradiction that the sequence converges. Suppose $\{x_k\}$ does not converge. Since there are only finite number of limit points to $\{x_k\}$, every limit point is an isolated one. Thus, Lemma 4.10 of [23] yields that there exists a subsequence $\{x_{k_j}\}$ of $\{x_k\}$ and an $\epsilon > 0$ such that $\|x_{k_j+1} - x_{k_j}\| \geq \epsilon$ for all j. Inequality (3.45) shows that $\|v_k\|$ tends to zero, which implies that $\|x_{k+1} - x_k\| \to 0$ because of (2.18). This contradiction shows that $\{x_k\}$ converges.

Note that θ_i goes to zero in criterion (2.14)-c). Since $\{x_k\}$ converges to a point x_*, from Lemma 3.6, we know that the second-order necessary optimality conditions are satisfied at x_*. □

4. Preliminary numerical experiments

In the final section, we present preliminary computational results to illustrate the performance of Algorithm 2.1. We test Algorithm 2.1 on a set of nonlinear equality constrained problems from the CUTE collection [2] and our own test problems. To exhibit the important role the local change of variables plays in Algorithm 2.1, we compare the implementations of Algorithm 2.1 to a variation of our algorithm without the change of variables. All our experiments were performed in MATLAB Version 4.1 on a Sun 4/670 workstation.

Table 1 gives a brief description of our test problem set. Most problems in Table 1 (all except TEST1 and TEST2) are selected from the CUTE collection [2]. These problems represent different sizes (in terms of m and n). Problems TEST1 and TEST2 are our own test

Table 1. Description of problems.

Problems		m	$nnz(A)$	Constraints
BT6	5	2	5	Nonlinear
BT11	5	3	8	Nonlinear
DTOC2	58	36	144	Nonlinear
DTOC4	29	18	65	Nonlinear
DTOC6	201	100	724	Nonlinear
GENHS28	300	298	894	Linear
MWRIGHT	5	3	8	Nonlinear
ORTHREGA	517	256	1792	Nonlinear
ORTHREGC	505	250	1750	Nonlinear
ORTHREGD	203	100	500	Nonlinear
TEST1	500	300	2308	Quadratic
TEST2	500	300	2661	Nonlinear

problems which we believe are hard problems. Problem TEST1 is to minimize a Rosenbrock function [12] subject to some quadratic equality constraints, i.e.,

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{n-1}\left[(1-x_i)^2+100\left(x_{i+1}-x_i^2\right)^2\right] \\ &\text{subject to} && a_i^T x+.5x^T M_i x=0, \quad i=1,\ldots,m \end{aligned} \tag{4.47}$$

where $a_i \in \Re^n$, $i = 1, 2, \ldots, m$, are vectors, and $M_i \in \Re^{n \times n}$, $i = 1, 2, \ldots, m$, are symmetric. To increase nonlinearity, in the TEST2 problems we add some perturbation functions to the objective function and the quadratic constraint functions in (4.47). Namely, we solve the following problem:

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{n-1}\left[(1-x_i)^2+100\left(x_{i+}-x_i^2\right)^2\right]+\delta_0(x) \\ &\text{subject to} && a_i^T x+.5x^T M_i x+\delta_i(x) \quad 0 \quad i=1,.\quad ,m \end{aligned} \tag{4.48}$$

where $\delta_0(x), \delta_i(x), i = 1, 2, \ldots, m$, are perturbation functions. The perturbation functions are generated randomly to be linear combinations of polynomials, trigonometry functions, logarithmic functions and exponential functions. For example, $\delta_0(x)$ could be

$$\begin{aligned} \delta_0(x) = &\left(x_1^2+x_4\right)^2+1.0+\log\left(1+x_2^2+x_3^2\right) \\ &+10\ \sin(2\pi x_5)\cos(2\pi x_6)-e^{-(x_5-x_3)^2}+ \end{aligned}$$

In problems TEST1 and TEST2, the matrices $[a_1, a_2, \quad, a_m]$ and $M_i, i = 1, 2, \ldots, m$, were created randomly.

Let d_{newton} denote an approximation (defined in (4.50)) to the Newton direction and d_{neg} denote a negative curvature direction. To reduce computational cost, we approximately solve subproblem $\min\{q_i^{(l)}(h) : \|h\|_2 \le \Delta_i^{(l)}\}$ for $Z_i^{(l+)}h$ by considering a 2-dimensional subproblem at each iteration:

$$\min\{q_i^{(l)}(h) : \|h\|_2 \le \Delta_i^{(l)}, h \in \text{span}\{g(x_i^{(l+)}), d\}\},$$

where

$$d = \begin{cases} d_{newton} & \text{if } H_i^{(l+)} \text{ is positive definite,} \\ d_{neg} & \text{otherwise} \end{cases}$$

$$\|H_i^{(l+)} d_{newton} + g(x_i^{(l+)})\} = \epsilon \|g(x_i^{(l+)})\|$$

and

$$d_{neg}^T H_i^{(l+)} d_{neg} \le \gamma \nu_{\min}(H_i^{(l+)}) \|d_{neg}\|_2^2.$$

In (4.51), $\gamma > 0$ is a constant and $\nu_{\min}(H_i^{(l+)})$ denotes the smallest eigenvalue of $H_i^{(l+)}$. In [5] it is proved that the solution to (4.49), $h = h_i^{(l)}$, satisfies conditions 1–3 stated in Section 2. In practice, inequality (4.51) is difficult to satisfy. We relax (4.51) and require only $d_{neg}^T H_i^{(l+)} d_{neg} \le 0$. However, this relaxation does have a (theoretical) cost; the resultant algorithm no longer guarantees convergence to a point satisfying the second-order necessary optimality conditions.

We compute the direction d by attempting the Cholesky factorization of $H(x_i^{(l+)})$. If $H(x_i^{(l+)})$ is positive definite, we solve (4.50) for an approximation to the Newton direction d_{newton}. A negative curvature direction d_{neg} is obtained when the Cholesky factorization fails (e.g., Section 4.4 in [17]). Such a direction d_{neg} satisfies $d_{neg}^T H_i^{(l+)} d_{neg} \le 0$.

We use the curve technique, $x_{k+1} = x_{k+} + s_{k+}(h)$, only when necessary. In particular, we take

$$x_{k+1} = x_{k+} + Z_{k+}h \tag{4.52}$$

when $r_k \ge \eta_1$. Otherwise, we use the curve technique to update the iterates

$$x_{k+1} = x_{k+} + s_{k+}(h),$$

where $s_{k+}(h) = Z_{k+}h + Y_{k+}R_{k+}^{-T}[c(x_{k+}) - c(x_{k+} + Z_{k+}h)]$.

When solving problems in Table 1 using Algorithm 2.1, we set the stopping criterion to be $\mu < \mu_{\min} = 10^{-7}$. We take $\mu_0 = \theta_0 = 1.0$, $\Delta_0 = \|Z(x_0)^T \nabla f(x_0)\|^{1/2}$, $\rho = 0.1$, and $\sigma = 0.0001$. For both TEST1 and TEST2, the starting points are $x_0 = [.5, .5, \ldots, .5]^T$. For the test problems drawn from the CUTE collection, we take the CUTE default values.

Table 2. Results using algorithm 2.1

Problems	Number of iterations	Function evaluations			*error*
		f, c	*g, A*	*H*	
BT6	10	69	20	10	$O(10^{-7})$
BT11	6	19	13	6	$O(10^{-9})$
DTOC2	12	32	17	9	$O(10^{-6})$
DTOC4	3	7	7	3	$O(10^{-6})$
DTOC6	50	126	59	50	$O(10^{-5})$
GENHS28	2	4	4	2	$O(10^{-14})$
MWRIGHT	8	20	16	8	$O(10^{-9})$
ORTHREGA	32	295	49	30	$O(10^{-8})$
ORTHREGC	8	24	17	8	$O(10^{-10})$
ORTHREGD	6	12	10	3	$O(10^{-8})$
TEST1	15	29	25	15	$O(10^{-5})$
TEST2	10	25	21	10	$O(10^{-7})$

Table 2 illustrates the results of our numerical experiments for problems in Table 1. The first column states the names of the problems. The second column shows the number of iterations needed to reach the stopping criteria. Column "function evaluations" gives the number of function evaluations needed for different problems. Sub-columns "f, c" indicate the number of function evaluations needed for the functions $f(x)$ and $c(x)$, "g, A" for their gradients, and "H" for the Hessians of the Lagrangian functions. On the sixth column the quantity "*error*" is defined as

$$error = \sqrt{\|Z(x)^T \nabla f(x)\|_2^2 + \|c(x)\|_2^2}, \tag{4.54}$$

where x is the approximate solution to (1.1).

We also solved the test problems from the CUTE collection in Table 1 using LANCELOT. When running LANCELOT, we used second derivatives and set the stopping criterion to be $error < 10^{-5}$ where *error* is defined in (4.54). We present the results of our LANCELOT experiments in Table 3.

Tables 2 and 3 suggest that Algorithm 2.1 is quite promising. We can see in Tables 2 and 3 that for most test problems Algorithm 2.1 takes fewer iterations and function evaluations than LANCELOT. For a few problems (BT6, DTOC2 and ORTHREGA) Algorithm 2.1 takes more function evaluations for "f" and "c" than LANCELOT. This is mainly due to the line search procedure in Algorithm 2.1. But for the same problems. (BT6, DTOC2 and ORTHREGA) Algorithm 2.1 takes fewer function evaluations for "g", "A" and "H" than LANCELOT.

Finally, we use problem DTOC6 as an example to illustrate the essential role the local change of variables plays in Algorithm 2.1. We use three variations of Algorithm 2.1 to solve problem DTOC6. In the first variation we perform the change of variables at every

Table 3. Results using LANCELOT.

Problems	Number of iterations	Function evaluations			*error*
		f, c	*g, A*	*H*	
BT6	28	28	25	28	$O(10^{-5})$
BT11	19	19	20	19	$O(10^{-6})$
DTOC2	18	18	18	18	$O(10^{-6})$
DTOC4	17	17	17	17	$O(10^{-7})$
DTOC6	59	59	60	59	$O(10^{-5})$
GENHS28	6	6	7	6	$O(10^{-7})$
MWRIGHT	19	19	19	19	$O(10^{-6})$
ORTHREGA	92	92	84	92	$O(10^{-5})$
ORTHREGC	30	30	28	30	$O(10^{-5})$
ORTHREGD	44	44	40	44	$O(10^{-5})$

Table 4. Comparison of different variations.

Variations	Number of iterations	Function evaluations			*error*
		f, c	*g, A*	*H*	
Variation 1	11	33	19	11	$O(10^{-7})$
Variation 2	50	126	59	50	$O(10^{-5})$
Variation 3	1170	1252	1182	1170	$O(10^{-4})$

iteration. The second variation is the same one we used to obtain results in Table 2: we try form (4.52) first and perform the change of variables when form (4.52) fails to provide good trust-region ratios. In the third variation we do not perform any change of variables at all and adopt form (4.52) for all iterations. The results are presented in Table 4.

Table 4 illustrates that the change of variables is significant to the performance of Algorithm 2.1. From Table 4 we can see that the first and second variations work well and the third one takes significantly more iterations and function evaluations to converge. Indeed, for almost all problems we tested, the third variation works poorly. Table 4 also indicates that the first variation is more efficient than the second for the particular problem DTOC6. However, for most problems in Table 1, the second variation is the best.

The change of variables does have a cost. This can be seen by comparing (4.52) and (4.53): an extra function evaluation and an extra solution of a triangular system. However, our numerical experiment indicates that the total number of iterations is often reduced, sometimes significantly. Therefore, the total number of function evaluations will also likely be reduced. The solution of a triangular system is usually cheap.

We do not clearly see superlinear convergence for most of our test problems, probably because we terminate the iteration before that occurs, i.e., μ_i is not sufficiently small yet. A difficulty is that when μ_i is too small, some calculations are numerically unstable.

Table 5. Results from [9]

Problems	Number of iterations	Function evaluations		*error*
		f, *c*	*g*, *A*	
BT6	12	37	21	$O(10^{-6})$
BT11	9	25	18	$O(10^{-7})$
DIPIGRI	16	70	27	$O(10^{-6})$
DTOC2	12	17	17	$O(10^{-5})$
DTOC4	4	8	8	$O(10^{-5})$
DTOC6	11	18	17	$O(10^{-6})$
GENHS28	6	9	8	$O(10^{-6})$
HS100	17	75	29	$O(10^{-7})$
MWRIGHT	14	36	22	$O(10^{-5})$
ORTHREGA	83	883	91	$O(10^{-5})$
ORTHREGC	24	61	31	$O(10^{-5})$
ORTHREGD	21	90	38	$O(10^{-5})$
TEST1	104	443	116	$O(10^{-5})$
TEST2	149	526	161	$O(10^{-5})$

Finally, it would be interesting to compare the performance of the new algorithm with that of the algorithm proposed in [9]. Table 5 is from [9] except that the test problems DIPIGRI and HS100 are not included here because we did not test them with the new algorithm (we already have a few such small sized problems). For most problems, the new algorithm takes fewer iterations and fewer function evaluations.

5. Concluding remarks

The quadratic penalty function is usually dismissed as a merit function for nonlinearly constrained minimization for two reasons: The Hessian matrix of the penalty function becomes singular as the solution is approached, and short steps must often be taken, when far from the solution, due to the effect of the penalty term. Coleman and Hempel [8] and Gould [8] have indicated how to circumvent potential problems due to Hessian singularity at the solution. In this paper we have proposed a new technique which not only avoids problems due to Hessian singularity at the solution, but also avoids excessively small steps when far from the solution.

The crux of our new approach is a local transformation, defined at each iteration, and the approximate solution of a trust-region problem in the new (local) variables which allows for the efficient computation of a 'large' curved step in the original variables. This curved step follows a quadratic approximation to the current constraint contour and therefore causes little increase in the penalty term.

We have established global convergence properties and demonstrated how this technique can be efficiently implemented. Finally, we have provided computational results of

preliminary numerical experiments. The computational and theoretical results indicate that this approach has considerable practical potential for solving nonlinear equality-constrained minimization problems.

Acknowledgments

We would like to thank Dr. Yuying Li for her helpful comments and suggestions on this paper. We thank the referees for an updated reference and for many helpful remarks and comments which improved the presentation of this paper.

This research was partially supported by the Applied Mathematical Science Research Program (KC-04-02) of the Office of Energy Research the U.S. Department of Energy under grant DE-FG02-86ER25013.A000.

References

1. P.T. Boggs, A.J. Kearsley, and J.W. Tolle, "A practical algorithm for general large scale nonlinear optimization problems," SIAM J. Optimization, vol. 9, pp. 755–778, 1999.
2. I. Bongartz, A.R. Conn, N.I.M. Gould, and Ph.L. Toint, "CUTE: Constrained and unconstrained testing environment," ACM Transactions on Mathematical Software, vol. 21, pp. 123–160, 1995.
3. R.H. Byrd and J. Nocedal, "An analysis of reduced Hessian methods for constrained optimization," Mathematical Programming, vol. 49, pp. 285–323, 1991.
4. R.H. Byrd, R.B. Schnabel, and G.A. Schultz, "A trust-region algorithm for nonlinearly constrained optimization," SIAM Journal on Numerical Analysis, vol. 24, pp. 1152–1170, 1987.
5. R.H. Byrd, R.B. Schnabel, and G.A. Schultz, "Approximate solution of the trust-region problem by minimization over two-dimensional subspaces," Mathematical Programming, vol. 40, pp. 247–263, 1988.
6. M.R. Celis, J.E. Dennis, and R.A. Tapia, "A trust-region strategy for nonlinear equality constrained optimization," in Numerical Optimization 1984. P.T. Boggs, R.H. Byrd, and R.B. Schnabel (Eds.), SIAM: Philadelphia, 1985, pp. 71–82.
7. T.F. Coleman, "On characterizations of superlinear convergence for constrained optimization," Lectures in Applied Mathematics, vol. 26, pp. 113–133, 1990.
8. T.F. Coleman and C. Hempel, "Computing a trust-region step for a penalty function," SIAM Journal on Scientific and Statistical Computing, vol. 11, pp. 180–201, 1990.
9. T.F. Coleman, J. Liu, and W. Yuan, "A quasi-Newton quadratic penalty method for minimization subject to nonlinear equality constraints," Computational Optimization and Applications, vol. 15, pp. 103–123, 2000.
10. T.F. Coleman and D. Sorensen, "A note on the computation of an orthonormal basis for the null space of a matrix," Mathematical Programming, vol. 29, pp. 234–242, 1984.
11. A.R. Conn, N.I.M. Gould, D. Orban, and Ph.L. Toint, "A primal-dual trust-region algorithm for non-convex nonlinear programming," Mathematical programming, vol. B87, pp. 215–249, 2000.
12. A.R. Conn, N.I.M. Gould, and Ph.L. Toint, "LANCELOT: A FORTRAN package for large-scale nonlinear optimization (Release A)," No. 17 in Springer Series in Computational Mathematics, Springer-Verlag: New York, 1992.
13. J.E. Dennis and R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall: Englewood Cliffs, New Jersey, 1983.
14. M. El-Alem, "A global convergence theory for the Celis-Dennis-Tapia algorithm for constrained optimization," Technical Report 88-19, Department of Mathematical Sciences, Rice University, Houston, TX, 1988.
15. R. Fletcher, Practical Methods of Optimization, 2nd edn., John Wiley and Sons: Chichester, 1987.
16. D.M. Gay, "Computing optimal locally constrained steps," SIAM Journal on Scientific and Statistical Computing vol. 2, pp. 186–197, 1981.

17. P.E. Gill, W. Murray, and M.H. Wright, Practical Optimization, Academic Press: London and New York, 1981.
18. N.I.M. Gould, "On the accurate determination of search directions for simple differentiable penalty functions," IMA Journal of Numerical Analysis, vol. 6, pp. 357–372, 1986.
19. N.I.M. Gould, "On the convergence of a sequential penalty function method for constrained minimization," SIAM J. On Numerical Analysis, vol. 26, pp. 107–128, 1989.
20. C.G. Hempel, "A globally convergent quadratic penalty function method with fast local convergence properties," Ph.D. Dissertation, Center for Applied Mathematics, Cornell University, 1990.
21. X. Liu and Y. Yuan, "A globally convergent, locally superlinearly convergent algorithm for equality constrained optimization," Research Report, ICM-97-84, Inst. Comp. Math. Sci./Eng. Computing, Chinese Academy of Sciences, Beijing, China.
22. N. Maratos, "Exact penalty function algorithms for finite dimensional and control optimization problems," Ph.D. thesis, University of London, 1978.
23. J.J. Moré and D.C. Sorensen, "Computing a trust-region step," SIAM Journal on Scientific and Statistical Computing, vol. 4, pp. 553–572, 1983.
24. E.O. Omojokun, "Trust-region algorithms for optimization with nonlinear equality and inequality constraints," Ph.D. thesis, University of Colorado, 1989.
25. M.J.D. Powell, "Convergence properties of a class of minimization algorithms," in Nonlinear Programming. O.L. Mangasarian, R.R. Meyer, and S.M. Robinson (Eds.), vol. 2, Academic Press: New York, 1975, pp. 1–27.
26. M.J.D. Powell and Y. Yuan, "A trust-region algorithm for equality constrained optimization," Mathematical Programming, vol. 49, pp. 189–211, 1991.
27. 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, vol. 22, pp. 47–67, 1985.
28. A. Vardi, "A trust-region algorithm for equality constrained minimization: Convergence properties and implementation," SIAM Journal of Numerical Analysis, vol. 22, pp. 575–591, 1985.