Convergence to a Second-Order Point of a Trust-Region Algorithm with a Nonmonotonic Penalty Parameter for Constrained Optimization

M. M. El-Alem

CRPC-TR96654
July 1996

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

Convergence to a Second-Order Point of a Trust-Region Algorithm with a Nonmonotonic Penalty Parameter for Constrained Optimization [1]

M. M. El-Alem [2]

Communicated by R. A. Tapia

To appear in JOTA
Volume 91, No. 1, October 1996.

[1]This research was supported by the Rice University Center for Research on Parallel Computation grant R31853 and the REDI Foundation.

[2]Associate Professor, Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt.

Abstract

In a recent paper, the author (Ref. 1) proposed a trust-region algorithm for solving the problem of minimizing a non-linear function subject to a set of equality constraints. The main feature of the algorithm is that the penalty parameter in the merit function can be decreased whenever it is warranted. He studied the behavior of the penalty parameter and proved several global and local convergence results. One of these results is that there exists a subsequence of the iterates generated by the algorithm, that converges to a point that satisfies the first-order necessary conditions.

In the current paper, we show that, for this algorithm, there exists a subsequence of iterates that converges to a point that satisfies both the first-order and the second-order necessary conditions.

Key Words : Constrained optimization, equality constrained, penalty parameter, non-monotonic penalty parameter, convergence, trust-region methods, first-order point, second-order point, necessary conditions.

1 Introduction

In this paper, we are interested in numerically approximating the solution of the following equality constrained optimization problem:

$$\begin{array}{lll} \text{(EQ)} & \min & f(x), \\ & \text{s. t.} & h(x) = 0, \end{array}$$

where $h(x) = [h_1(x), ..., h_m(x)]^T$. The functions f and $h_i, i = 1, 2, ..., m$ are assumed to be at least twice continuously differentiable and the matrix $\nabla h(x) = [\nabla h_1(x), ..., \nabla h_m(x)]$ is assumed to have full column rank at every x in the range of interest.

The Lagrangian function associated with problem (EQ) is the function $\ell(x, \lambda) = f(x) + \lambda^T h(x)$ where $\lambda \in \Re^m$ is the Lagrange multiplier vector.

First and second order optimality conditions can be stated in terms of the Lagrangian function as follows. The first-order necessary conditions for a point $x_\star \in \Re^n$ to be a solution of problem (EQ) is feasibility, $h(x_\star) = 0$, and the existence of a Lagrange multiplier vector $\lambda_\star \in \Re^m$ such that the point $(x_\star, \lambda_\star)$ satisfies $\nabla_x \ell(x_\star, \lambda_\star) = 0$. *i. e.*, the point $(x_\star, \lambda_\star)$ is a solution to the following $(n + m) \times (n + m)$ nonlinear system of equations:

$$\begin{bmatrix} \nabla_x l(x, \lambda) \\ h(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{1}$$

If in addition, the matrix $\nabla_x^2 \ell(x_\star, \lambda_\star)$ is positive semi-definite on the null space of $\nabla h(x_\star)^T$, then we say that the point $(x_\star, \lambda_\star)$ satisfies the second-order necessary conditions.

The point $(x_\star, \lambda_\star)$ is said to satisfy the second-order sufficiency conditions if it satisfies the first-order necessary conditions and the matrix $\nabla_x^2 \ell(x_\star, \lambda_\star)$ is positive definite on the null space of $\nabla h(x_\star)^T$. For a detailed discussion of the optimality conditions, see, for example, Ref. 2.

Trust-region algorithms for the equality constrained optimization problem are a class of numerical algorithms for finding an approximate solution to problem (EQ). They are iterative methods that compute, at every iteration k, a trial step s_k by solving a trust-region subproblem. The step s_k is then tested using a merit function. Such merit function often involves a parameter that is called the penalty parameter. This parameter is updated using an updating scheme. The step is accepted only if x_{k+1} is a better approximation to the solution than x_k and the radius of the trust region is updated accordingly. The aim is to allow convergence to a solution from any starting point. This approach has proven to be very successful both theoretically and practically.

The rest of this section contains historical background. In Section 1.1, a survey on some local methods for solving problem (EQ) is presented. A brief survey of trust-region methods for equality constrained optimization is presented in Section 1.2. For a detailed discussion about trust-region methods for problem (EQ), see Ref. 3.

The rest of the paper is organized as follows. In Section 2, we present the trust-region algorithm that was proposed by the author, Ref. 1. In Section 3, we state the assumptions that are needed to establish our convergence result. In Section 4, we present the convergence result. We start with some intermediate lemmas that are needed in the proof of our result. We then present our main convergence result. Section 5 contains concluding remarks.

Throughout this paper, all the norms used are l_2 norms and subscripted values of functions are used to denote evaluation at a particular point. For example f_k means $f(x_k)$, ℓ_k means $\ell(x_k, \lambda_k)$, and so on.

1.1 Local Method

Newton's method is known to be an effective local method for finding a root of nonlinear system of equations. Under mild assumptions, it possesses fast local convergence. A natural way to obtain a local method for solving problem (EQ) is to use Newton method to find a root of the nonlinear system of equations (1). This gives rise to the following $(n+m)\times(n+m)$ linear system that has to be solved at the k-th iteration.

$$\begin{bmatrix} {\nabla_x}^2 l_k & \nabla h_k \\ \nabla h_k^T & 0 \end{bmatrix} \begin{bmatrix} s_k \\ \Delta\lambda_k \end{bmatrix} = - \begin{bmatrix} \nabla_x l_k \\ h_k \end{bmatrix}. \tag{2}$$

The solution obtained by the above method satisfies the first-order necessary conditions (1). However, it need not necessarily satisfy the second-order necessary conditions of problem (EQ). This implies that, the computed solution obtained from the above method need not necessarily be a minimizer of problem (EQ).

The successive quadratic programming method is a method that has the flavor of the original problem in the sense that it solves a minimization problem at every iteration. In particular, at each iteration, the SQP method obtains a step s_k^{QP} and an associated Lagrange multiplier step $\Delta\lambda_k^{QP}$ by solving the following quadratic programming problem.

$$\begin{array}{lll} \text{(QP)} & \min & \nabla_x \ell_k^T s + \frac{1}{2} s^T \nabla_x^2 \ell_k s, \\ & \text{s. t.} & h_k + \nabla h_k^T s = 0. \end{array}$$

This method is one of the most popular and successful methods for solving problem (EQ). The basic idea of the SQP method probably goes back to the beginning of this century. However, the earliest published reference to the SQP method that we are aware of is Ref. 4.

If the solution to problem (QP) exists, then the SQP method is equivalent to Newton's method and therefore possesses fast local convergence. However, because the second-order sufficiency conditions need not hold at each iteration, there is a fundamental difficulty in the definition of the SQP step. By this we mean that, at each iteration k, the matrix $\nabla_x^2 \ell_k$ need not be positive definite on the null space of ∇h_k^T; hence the (QP) subproblem may not have a solution. Even if we assume that problem (QP) can be solved at every iteration, there is no guarantee that the sequence generated by this method will converge. See Ref. 5 for more details.

These difficulties imply that the SQP algorithm can not be guaranteed to work without a modification. An effective modification that deals with the lack of positive definiteness on the null space and guarantees global convergence is the use of a trust-region globalization strategy. This takes us to the following section.

1.2 Trust-Region Algorithms for Problem (EQ)

As an attempt to avoid the difficulties discussed above, a trust-region globalization strategy is incorporated into the SQP method. The aim is to avoid the fundamental difficulty in trying to ensure that the (QP) subproblem has a solution at every iteration. This is done by restricting the size of the step to a region where the model subproblem can be trusted. This added constraint causes the region of interest to be bounded and therefore the subproblem is guaranteed to have a solution regardless of the nature of the matrix $\nabla_x^2 \ell$, provided that the feasible region is not empty.

If we add a trust-region constraint to the (QP) subproblem in a straightforward manor, we obtain the following trust-region subproblem

$$\text{(TRQP)} \qquad \begin{array}{ll} \min & \nabla_x \ell_k^T s + \frac{1}{2} s^T \nabla_x^2 \ell_k s, \\ \text{s. t.} & h_k + \nabla h_k^T s = 0, \\ & \|s\| \leq \Delta_k. \end{array}$$

However, the two constraints may be inconsistent in the sense that the hyperplane $h_k + \nabla h_k^T s = 0$ and the trust-region ball may be disjoint and therefore this approach may lead to an infeasible subproblem. To avoid this difficulty several approaches have been investigated.

The first approach is to relax the linear constraints in such a way that the resulting feasible set is non-empty. In this approach the linear constraints in (TRQP) is replaced by the following relaxed constraints

$$\alpha_k h_k + \nabla h_k^T s = 0,$$

where $\alpha_k \in [0,1]$. This approach was suggested in Ref. 6 and was used in Refs. 7-8. A major difficulty with this approach lies in the problem of choosing α_k so that a feasible trust-region subproblem is ensured.

The second approach is to replace the linear constraints in (TRQP) by

$$\|h_k + \nabla h_k^T s\|^2 \leq \theta_k^2.$$

If θ_k is chosen properly, the resulting subproblem is always feasible. This approach was suggested in Ref. 9 and was used in Refs. 10-12. The parameter θ_k is chosen to ensure a sufficient decrease in the quadratic model of the linearized constraints. This decrease is at least a fraction of the decrease obtained by the Cauchy point. The Cauchy point is defined to be the optimal point inside the trust region in the steepest descent direction for the function $\|h_k + \nabla h_k^T s\|^2$. That is, the minimum point inside the trust region in the steepest descent direction. In Refs. 10-11, the parameter θ_k was taken to be

$$\theta_k^2 = (1-r)\|h_k\|^2 + r\|h_k + \nabla h_k^T s^{cp}\|^2,$$

for some fixed $r \in (0,1)$, where s^{cp} is the step to the Cauchy point. In Ref. 12, the choice of θ_k was

$$\theta_k^2 = \{\min \|h_k + \nabla h_k^T s\|^2 \ : \ \sigma_1 \Delta_k \leq \|s\| \leq \sigma_2 \Delta_k\},$$

where $0 < \sigma_1 \leq \sigma_2 \leq 1$.

A major disadvantage with this approach lies in the fact that the resulting trust-region subproblem has two quadratic constraints and there is no efficient algorithm for finding a good approximation to the solution of this subproblem. See Refs. 13-17 for algorithms that were suggested to solve special cases of this subproblem.

The reduced Hessian technique is another approach to overcome the difficulty of having an infeasible trust-region subproblem. In this approach, the step is decomposed into two components; the tangential component s^t and the normal component s^n. The step s^n is computed by solving a trust-region subproblem. The tangential component s^t is then obtained by solving another trust-region subproblem. This approach has been used in Refs. 1,

3, 18-25. One of the advantages of this approach is that the two subproblems that we have to solve at every iteration are similar to the trust-region subproblem for the unconstrained case. Our way of computing the trial steps uses this approach and is presented in detail in the following section.

2 Trust-Region Algorithm

In this section we present the algorithm. In Section 2.1, we present the trial step computation strategy. In Section 2.2, we present the trial-step acceptance mechanism and the trust-region updating rules. Section 2.3 is devoted to presenting the penalty parameter updating scheme. Finally, in Section 2.4 we present an overall summary of the algorithm.

2.1 Computing the Trial Steps

At each iteration k, a trial step s_k is computed. Let s_k be decomposed into two orthogonal components; the normal component s_k^n and the tangential component $Z_k v_k$, where $v_k \in \Re^{m-n}$ and Z_k is an $n \times (n-m)$ matrix that forms an orthonormal basis for the null space of ∇h_k^T. The trial step has the form

$$s_k = s_k^n + Z_k v_k,$$

The matrix Z_k is obtained from the QR factorization of ∇h_k as follows

$$\nabla h_k = \left[\begin{array}{cc} Y_k & Z_k \end{array}\right] \left[\begin{array}{c} R_k \\ 0 \end{array}\right]. \tag{3}$$

To compute a trial step, two model trust-region subproblems are solved for s_k^n and v_k. We start by solving for s_k^n the following trust-region subproblem

$$\begin{array}{ll} \min & \|h_k + \nabla h_k^T s^n\|, \\ \text{s. t.} & \|s^n\| \leq \Delta_k. \end{array}$$

To obtain the tangential component, we solve for v_k the following trust-region subproblem

$$\min \quad (Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k + \tfrac{1}{2} {v_k}^T Z_k^T B_k Z_k v_k, \tag{4}$$

$$\text{s. t.} \quad \|Z_k v_k\| \leq \Delta_k, \tag{5}$$

where B_k is the Hessian of the Lagrangian $\nabla_x^2 \ell_k$.

Once the trial step is computed, it needs to be tested to determine whether it will be accepted. To do that, an estimate for the Lagrange multiplier λ_{k+1} is needed. We compute λ_{k+1} by solving the following least-squares problem:

$$\lambda_{k+1} = \text{argmin} \|\nabla h_{k+1} \lambda + \nabla f_{k+1}\|. \tag{6}$$

Using (3), solving (6) is equivalent to solving

$$R_{k+1} \lambda_{k+1} = -Y_{k+1}^T \nabla f_{k+1}. \tag{7}$$

2.2 Testing the Steps

Let s_k be the step computed by the algorithm and let λ_{k+1} be the Lagrange multiplier obtained by solving (7). We test whether the point $(x_k + s_k, \lambda_{k+1})$ will be taken as a next iterate. In order to do this, Fletcher's exact penalty function is employed as a merit function. It is the function

$$\Phi(x, \lambda, r) = f(x) + \lambda(x)^T h(x) + r\|h(x)\|^2,$$

where $\lambda(x)$ is the least-squares estimate of the multiplier discussed above and r is a penalty parameter. This function has been used as a merit function in trust-region algorithms in Refs. 1, 12.

We define the actual reduction in the merit function in moving from (x_k, λ_k) to $(x_k + s_k, \lambda_{k+1})$ to be

$$A_k = \Phi(x_k, \lambda_k, r_k) - \Phi(x_k + s_k, \lambda_{k+1}, r_k).$$

This can be written as

$$A_k = \ell(x_k, \lambda_k) - \ell(x_k + s_k, \lambda_k) - (\lambda_{k+1} - \lambda_k)^T h(x_k + s_k) + r_k[\ \|h_k\|^2 - \|h(x_k + s_k)\|^2].$$

The predicted reduction that we use has the form:

$$\begin{aligned} P_k \quad &= \quad -\nabla_x \ell_k^T s_k - \frac{1}{2} {s_k}^T B_k Z_k v_k - (\lambda_{k+1} - \lambda_k)^T [h_k + \frac{1}{2} \nabla h_k^T s_k] \\ &\quad\quad + r_k[\ \|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2]. \end{aligned}$$

This form of predicted reduction was first suggested in Ref. 12.

As in Ref. 1, the normal predicted decrease N_k is defined to be the decrease at the $k^{\underline{th}}$ iteration in the linearized model of the constraints by the step s_k^n. It predicts the actual

reduction in the violation of the constraints obtained by the normal component s_k^n and is given by:

$$N_k = \|h_k\|^2 - \|h_k + \nabla h_k^T s_k^n\|^2.$$

The tangential predicted decrease T_k is defined to be the decrease at the $k^{\underline{th}}$ iteration by the step $Z_k v_k$ in the quadratic model of the Lagrangian restricted to the null space of ∇h_k^T. It predicts the actual reduction in the Lagrangian function obtained by the tangential component $Z_k v_k$. It is given by:

$$T_k = -(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k - \frac{1}{2} v_k^T Z_k^T B_k Z_k v_k \ .$$

The acceptable step should be the step that produces a decrease in the merit function Φ. To test for this, the predicted reduction has to be made greater than zero. This is done first by increasing the penalty parameter if necessary. We will discuss this issue in the next section. But assume for the moment that $P_k > 0$. The trial steps are then tested and are accepted only if the actual reduction is greater than some fraction of the predicted reduction. Define

$$\Gamma_k = \frac{A_k}{P_k}.$$

The step s_k is accepted if $\Gamma_k \geq \eta_1$ where $\eta_1 \in (0,1)$. Typically, the value of the constant η_1 is taken to be very small. *e. g.* $\eta_1 = 10^{-4}$. If the step is judged acceptable, then we proceed to the next iteration. Otherwise, the trial step is rejected, the trust-region radius is decreased, and another trial step is computed from x_k in the smaller trust-region. *i. e.*, the index k is increased only if the step is accepted.

We reject the step if $\Gamma_k < \eta_1$. In this case, we decrease the radius of the trust region by picking $\Delta_k \in [a_1\|s_k\|, a_2\|s_k\|]$, where $0 < a_1 \leq a_2 < \frac{1}{\sqrt{2}}$.

Let $\eta_1 < \eta_2 < \eta_3 < \eta_4 < 1$. If the step is accepted, then the trust-region radius is updated as follows. If $\eta_2 \leq \Gamma_k < \eta_3$ then the radius of the trust region is kept the same. If the agreement between the actual reduction and the predicted reduction is poor; $\Gamma_k < \eta_2$, then we allow possible reduction in the radius of the trust region. We set $\Delta_{k+1} = \min(\Delta_k, a_2\|s_k\|)$. If on the other hand, the agreement between the actual reduction and the predicted reduction is fair, $\eta_3 \leq \Gamma_k < \eta_4$, then possibly increase the trust region. Set $\Delta_{k+1} = \min\{\Delta_\star, \max(\Delta_k, a_2\|s_k\|)\}$, where $\Delta_\star$ is an upper bound on the trust-region radius. If the agreement between the actual reduction and the predicted reduction is good, $\Gamma_k \geq \eta_4$, then, unless $\Delta_k = \Delta_\star$, increase the trust-region radius. Set $\Delta_{k+1} = \min\{\Delta_\star, \max(a_2\Delta_k, a_3\|s_k\|)\}$, where $a_3 > \frac{1}{\sqrt{2}}$. Note that for all k, we have $\Delta_k \leq \Delta_\star$.

We conclude this subsection by saying that the above represents our practical way of updating the trust-region radius. Our convergence analysis that will be presented in Section 4 only requires that the trust-region radius be decreased when the step is rejected. *i. e.* when $\Gamma_k < \eta_1$. On the other hand, when $\Gamma_k \geq \eta_1$, it requires that the trust-region radius be increased or kept the same. The analysis also requires that, for all k, $\Delta_k \leq \Delta_\star$.

2.3 Updating the Penalty Parameter

In Ref. 1, the author has proposed a scheme for updating the penalty parameter that allows it to be decreased whenever it is warranted. He proved several global and local convergence results. The theory allows the sequence $\{r_k\}$ to be non-monotonic. However, it requires that, for all k, $\underline{\rho}_{k-1} \leq r_k$, where $\underline{\rho}_k$ is defined below. This scheme can be stated as follows:

Scheme 2.1 Updating the Penalty Parameter

Given a constant $\rho > 0$ and an integer $N > 0$:

Set $r_o = r_{-1} = ... = r_{-(N-1)} = 1$

At each iteration k, do

Find
$$\begin{aligned} \underline{\rho}_{k-1} &= \min\{r_{k-1}, r_{k-2}, ..., r_{k-N}\}, \\ \overline{\rho}_{k-1} &= \max\{r_{k-1}, r_{k-2}, ..., r_{k-N}\}. \end{aligned}$$

Set
$$\rho_{k-1} = \min\{ \ \underline{\rho}_{k-1} + \rho \ , \ \overline{\rho}_{k-1} \ \}.$$

Set $r_k = \rho_{k-1}$.

If
$$P_k < \frac{\rho_{k-1}}{2}[\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2],$$

then set
$$r_k = 2\left\{ \frac{\nabla_x \ell_k^T s_k + \frac{1}{2} s_k^T B_k Z_k v_k + (\lambda_{k+1} - \lambda_k)^T [h_k + \frac{1}{2} \nabla h_k^T s_k]}{\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2} \right\} + \rho \ .$$

One positive feature of this scheme is that if at any iteration k we have $\Gamma_k < \eta_1$, then we reject the trial step and do not increase the iteration count k. As a consequence the set $\{r_{k-1}, \cdots, r_{k-N}\}$ remains unchanged. Thus, implicitly, the value of the penalty parameter is rejected and the only effect that an unacceptable trial step has is a decrease in the trust-region radius.

Another advantage of this scheme is that it does not enter in the step calculation, although it does enter into the process of testing the steps.

Also, the way of updating the penalty parameter ensures a predicted decrease in the merit function given by:

$$P_k \geq \frac{r_k}{2}[\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2].$$

That is, the predicted decrease is at least as much as the decrease in the linearized model of the constraints obtained by the normal component of s_k. So, at each iteration k, we have:

$$P_k \geq \frac{r_k}{2} N_k \ . \tag{8}$$

Finally, for updating the matrix B_k, the exact Hessian is used. *i. e.*, at each iteration k, we compute $B_k = \nabla_x^2 \ell_k = \nabla^2 f_k + \nabla^2 h_k \lambda_k$.

2.4 Summary of the Algorithm

Putting the pieces together, we can now outline the trust-region algorithm for finding a local minimizer of problem (EQ).

Algorithm 2.1 The Trust-Region Algorithm:

Initialization: Choose $x_0 \in \Re^n$ *and* $\lambda_0 \in \Re^m$. *Compute* $B_0 = \nabla_x^2 \ell(x_0, \lambda_0)$. *Set* $k = 0$.

At every iteration, do the following steps:

Step 1. Check for convergence.

Step 2. Compute s_k, λ_{k+1} *according to Section 2.1.*

Step 3. Update the penalty parameter according to Scheme 2.1.

Step 4. Test the step and update Δ_k *as in Section 2.2.*

Step 5. Compute $B_{k+1} = \nabla_x^2 \ell_{k+1}$.

Step 6. Set $k := k + 1$.

3 Assumptions

Let the sequence of iterates generated by the algorithm be $\{x_k\}$, and let $\Omega \in \Re^n$ be a convex set such that for all k, x_k and $x_k + s_k \in \Omega$, where s_k represents all the trial steps computed at iteration k. For such a set we assume,

(A1) f and $h_i \in C^2(\Omega)$ $i = 1, ..., m$.

(A2) $\nabla h(x)$ has full column rank for all $x \in \Omega$.

(A3) $f(x)$, $h(x)$, $\nabla h(x)$, $\nabla f(x)$, $\nabla^2 f(x)$, $R(x)^{-1}$ and each $\nabla^2 h_i(x)$, for $i = 1, ..., m$ are all uniformly bounded in Ω.

(A4) $\nabla^2 f$ and $\nabla^2 h_i$, $i = 1, ..., m$ are Lipschitz continuous in Ω.

An immediate consequence of the above assumptions is the existence of a constant $b > 0$, such that, for all k,

$$\|B_k\| \leq b, \quad \|Z_k^T B_k Z_k\| \leq b, \text{ and } \|Z_k^T B_k\| \leq b. \tag{9}$$

Another immediate consequence of these assumptions is the existence of constants $b_0 > 0$ and $b_1 > 0$ such that, for all k,

$$\|s_k^n\| \leq b_0 \|h_k\| \tag{10}$$

and

$$\|\lambda_{k+1} - \lambda_k\| \leq b_1 \|s_k\|. \tag{11}$$

The above assumptions are standard in the sense that they have been used by many authors. See, for example, Refs. 1, 3, 7-8, 10-12, 18-19, 23, 25.

4 Convergence Results

In this section we present our convergence result. We start by stating some intermediate lemmas. These lemmas are needed in the proof of our main result which will be presented in Section 4.2.

4.1 Intermediate Results

In this subsection we present some lemmas needed in the proof of the main result.

The following lemma shows that, at any iteration k, the normal predicted reduction N_k is at least equal to the decrease in the l_2 norm of the linearized constraints obtained by the Cauchy step. *i. e.* it satisfies a fraction of the Cauchy decrease condition.

Lemma 4.1. Assume (A1)-(A3). Then at any iteration k, N_k satisfies

$$N_k \geq b_2 \|h_k\| \min[b_3 \|h_k\|, \Delta_k], \tag{12}$$

where b_2 and b_3 are positive constants that do not depend on k.

Proof. The proof is similar to the proof of Lemma 6.1, Ref. 10. □

The following lemma shows that the tangential predicted reduction T_k is at least equal to the decrease in the quadratic model of the Lagrangian obtained by the Cauchy step. *i. e.* it satisfies a fraction of Cauchy decrease condition.

Lemma 4.2. Assume (A1)-(A3). Then for all k, the tangential predicted reduction satisfies:

$$T_k \geq \frac{1}{4}\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \min[\Delta_k, \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{2b}] . \tag{13}$$

Proof. For a proof see Lemma 3.2, Ref. 11. □

Let γ_k be the Lagrange multiplier of the trust-region constraint in the trust-region subproblem (4)-(5), then the following lemma gives a lower bound to the tangential predicted reduction T_k in terms of γ_k and Δ_k. It shows that T_k is at least equal to a fraction of the decrease in the quadratic model of the Lagrangian obtained by the optimal step. *i. e.* it satisfies a fraction of the optimal decrease condition. See Ref. 26 for more details about the optimal decrease.

Lemma 4.3. Assume (A1) and (A3). Then for all k, the tangential predicted reduction satisfies:

$$T_k \geq b_4 \gamma_k \Delta_k^2, \tag{14}$$

where b_4 is a positive constant that does not depend on k.

Proof. See Ref. 26. See also Theorem 4.17, Ref. 27. □

The following lemma gives a relation between the predicted decrease P_k and both the tangential and the normal predicted decrease, T_k and N_k.

Lemma 4.4. Assume (A1)-(A3). Then for all k, there exists a positive constant b_5, that does not depend on k, such that

$$P_k \geq T_k - b_5\|s_k\|\|h_k\| + \frac{r_k}{2} N_k. \tag{15}$$

Proof. See the proof of Lemma 5.4, Ref. 1. □

The following two lemmas give bounds on how accurate our notion of predicted reduction is as an approximation to the actual reduction.

Lemma 4.5. Assume (A1)-(A3). Then for any $x_k,\ x_k + s_k \in \Omega$, we have

$$|A_k - P_k| \leq b_6 r_k \Delta_k^2, \tag{16}$$

where b_6 is a positive constant independent of k.

Proof. The proof is similar to the proof of Corollary 6.4, Ref. 11. □

Lemma 4.6. Assume (A1)-(A4). Then for any $x_k,\ x_k + s_k \in \Omega$, we have

$$|A_k - P_k| \leq r_k(b_7\Delta_k + b_8\|h_k\|)\Delta_k^2, \tag{17}$$

where b_7 and b_8 are positive constants independent of k.

Proof. The proof is similar to the proof of Lemma 6.3, Ref. 11. □

4.2 Main Result

In this section, we present our main result. It says that the sequence of iterates generated by the algorithm will not be bounded away from a point that satisfies the second order necessary condition. In other words, there exists a subsequence of the sequence of iterates generated by the algorithm that will converge to a stationary point that satisfies the second order necessary condition. We start with the following asymptotic result.

Theorem 4.1. Assume (A1)-(A4). Then the sequence of iterates generated by the algorithm satisfies

$$\liminf_{k\to\infty} \left[\|Z_k^T\nabla f(x_k)\| + \|h(x_k)\| + \gamma_k\right] = 0.$$

Proof. The proof is by contradiction. Suppose that the above limit does not hold. Then there exists a constant $\varepsilon > 0$ such that for all k

$$\|Z_k^T\nabla f(x_k)\| + \|h(x_k)\| + \gamma_k > \varepsilon. \tag{18}$$

First we show that under (18) at any given iteration k, the next iteration $k+1$ can be computed. In other words, at a given point x_k an acceptable step must be found. *i. e.* the algorithm can not loop infinitely without finding an acceptable step.

If $h_k \neq 0$, then using (8) and Lemmas 4.1 and 4.5, we can write

$$|\Gamma_k - 1| \ = \ \left|\frac{A_k - P_k}{P_k}\right| \leq \frac{b_6\Delta_k^2}{b_2\|h_k\|\min\{b_3\|h_k\|, \Delta_k\}}.$$

Now as Δ_k gets small the quantity $|\Gamma_k - 1|$ approaches zero and an acceptable step must be found. On the other hand, if $h_k = 0$, and assume for the moment that at the point x_k the value of the penalty parameter r_k is finite (we will see next that this is the case for all k), then using (18) either $\|Z_k^T\nabla f(x_k)\| > \frac{\varepsilon}{2}$ or $\gamma_k > \frac{\varepsilon}{2}$. Consider the first case and use Lemmas 4.2, 4.4, and 4.5, we obtain

$$|\Gamma_k - 1| \ = \ \left|\frac{A_k - P_k}{P_k}\right| \leq \frac{b_6 r_k\Delta_k^2}{\frac{\varepsilon}{8}\min\{\Delta_k, \frac{\varepsilon}{4b}\}}.$$

Now as Δ_k gets small the quantity $|\Gamma_k - 1|$ approaches zero and an acceptable step must be found. Note that when $h_k = 0$, Lemma 4.4 implies that

$$P_k \geq T_k + \frac{r_k}{2} N_k$$

and no need to increase the penalty parameter. So at x_k the value of the penalty parameter will remain the same through all these unacceptable trial steps. Consider the case when $\gamma_k > \frac{\varepsilon}{2}$. In this case using Lemmas 4.3, 4.4, and 4.6, we have

$$|\Gamma_k - 1| \; = \; \left| \frac{A_k - P_k}{P_k} \right| \leq \frac{b_7 r_k \Delta_k}{\frac{\varepsilon}{2} b_4},$$

and again as Δ_k gets small the quantity $|\Gamma_k - 1|$ approaches zero and an acceptable step must be found.

We then show that under (18) the penalty parameter will reach an upper bound and will remain unchanged. To show this we consider two cases for $\|h_k\|$. Consider first the case when $\|h_k\| \leq c_1 \Delta_k$, where $c_1 > 0$ is a small constant that satisfies:

$$c_1 \leq \min \left\{ \frac{\varepsilon}{3\Delta_\star}, \frac{\varepsilon}{6bb_0\Delta_\star}, \frac{\varepsilon}{48\sqrt{2}b_5\Delta_\star} \min(1, \frac{\varepsilon}{12b\Delta_\star}), \frac{\varepsilon b_4}{6\sqrt{2}b_5} \right\}. \tag{19}$$

Since $c_1 \leq \frac{\varepsilon}{3\Delta_\star}$, then $\|h_k\| \leq \frac{\varepsilon}{3}$ and because of (18), $\|Z_k^T \nabla f_k\| + \gamma_k > \frac{2\varepsilon}{3}$. There are two cases to consider. First, if $\|Z_k^T \nabla f_k\| > \frac{\varepsilon}{3}$, we obtain

$$\begin{aligned} \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| &\geq \|Z_k^T \nabla f_k\| - \|Z_k^T B_k\| \|s_k^n\| \\ &\geq \frac{\varepsilon}{3} - bb_0 \|h_k\| \geq \frac{\varepsilon}{3} - \frac{\varepsilon}{6} = \frac{\varepsilon}{6}. \end{aligned}$$

Hence, using Lemmas 4.2, 4.4, and the fact that $\|s_k\| \leq \sqrt{2}\Delta_k$, we obtain

$$\begin{aligned} P_k &\geq \frac{1}{2} T_k + \frac{\varepsilon \Delta_k}{48} \min[1, \frac{\varepsilon}{12b\Delta_\star}] - \sqrt{2} c_1 b_5 \Delta_k^2 + \frac{r_k}{2} N_k \\ &\geq \frac{1}{2} T_k + \{\frac{\varepsilon}{48} \min[1, \frac{\varepsilon}{12b\Delta_\star}] \;-\; \sqrt{2} c_1 b_5 \Delta_\star\} \Delta_k + \frac{r_k}{2} N_k. \end{aligned}$$

From (19), the quantity $\{\frac{\varepsilon}{48} \min[1, \frac{\varepsilon}{12b\Delta_\star}] \;-\; \sqrt{2} c_1 b_5 \Delta_\star\}$ is positive. Hence,

$$P_k \geq \frac{1}{2} T_k + \frac{r_k}{2} N_k.$$

On the other hand if $\gamma_k > \frac{\varepsilon}{3}$, then using Lemmas 4.3 and 4.4, we obtain

$$\begin{aligned} P_k &\geq \frac{1}{2} T_k + \frac{\varepsilon b_4}{6} \Delta_k^2 - \sqrt{2} c_1 b_5 \Delta_k^2 + \frac{r_k}{2} N_k \\ &\geq \frac{1}{2} T_k + \{\frac{\varepsilon b_4}{6} \;-\; \sqrt{2} c_1 b_5\} \Delta_k^2 + \frac{r_k}{2} N_k. \end{aligned}$$

From (19), the quantity $\{\frac{\varepsilon b_4}{6} - \sqrt{2}c_1 b_5\}$ is positive. Hence, we have

$$P_k \geq \frac{1}{2}T_k + \frac{r_k}{2}N_k.$$

So, in the case when $\|h_k\| \leq c_1\Delta_k$, where c_1 is as in (19), there is no need to increase the penalty parameter. Consider now the second case when $\|h_k\| > c_1\Delta_k$. This is the only case where r_k may be increased. But if we follow a proof similar to the proof of Lemma 5.8 Ref. 1, we demonstrate the boundedness of the sequence $\bar{\rho}_k$. The proof of the boundedness of the sequence $\{r_k\}$ follows from the fact that for all k, $r_k \leq \bar{\rho}_k$. The fact that when the penalty parameter is increased it will be increased by at least ρ implies that there exists a constant k_0 such that for all $k \geq k_0$, $r_k = r_{k_0}$.

We can now use argument similar to Lemma 5.1, Ref. 11 and conclude that at any iteration k, the algorithm can not loop infinitely without finding an acceptable step. This result allows us to drop the consideration of the trial steps and only consider successful steps.

At this point, the convergence of h_k to zero is evident using the same proof as in Lemma 5.12, Ref. 1.

Now since h_k converges to zero, there exists k_1 sufficiently large such that for all $k \geq k_1$, we have

$$\|h_k\| < \min\left\{\frac{\varepsilon}{3}, \frac{\varepsilon}{6bb_0}, \frac{\varepsilon}{48\sqrt{2}b_5}\min[1, \frac{\varepsilon}{12b\Delta_\star}]\right\}. \tag{20}$$

But using (18) inequality (20) will imply that for all $k \geq k_1$, $\|Z_k^T \nabla f_k\| + \gamma_k \geq \frac{2\varepsilon}{3}$. There are two cases to consider. First, if at any iteration $k > k_1$, we have $\|Z_k^T \nabla f_k\| > \frac{\varepsilon}{3}$, then

$$\begin{aligned} \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| &\geq \|Z_k^T \nabla f_k\| - \|Z_k^T B_k\| \|s_k^n\| \\ &\geq \frac{\varepsilon}{3} - bb_0\|h_k\| \geq \frac{\varepsilon}{3} - \frac{\varepsilon}{6} = \frac{\varepsilon}{6}. \end{aligned}$$

Hence, using $\|s_k\| \leq \sqrt{2}\Delta_k$, we have

$$P_k \geq \frac{1}{2}T_k + \{\frac{\varepsilon}{48}\min[1, \frac{\varepsilon}{12b\Delta_\star}] - \sqrt{2}b_5\|h_k\|\}\Delta_k + \frac{r_k}{2}N_k.$$

Now using (20), we obtain

$$P_k \geq \frac{1}{2}T_k + \frac{r_k}{2}N_k > \{\frac{\varepsilon}{48\Delta_\star}\min[1, \frac{\varepsilon}{12b\Delta_\star}]\}\Delta_k^2.$$

On the other hand, if at any iteration $k \geq k_1$, we have $\gamma_k > \frac{\varepsilon}{3}$, we have

$$P_k \geq \frac{1}{2}T_k + \{\frac{\varepsilon b_4}{6}\Delta_k - \sqrt{2}b_5\|h_k\|\}\Delta_k + \frac{r_k}{2}N_k.$$

If at this iteration $\|h_k\| \le \frac{\varepsilon b_4 \Delta_k}{6\sqrt{2}b_5}$, then

$$P_k \ge \frac{\varepsilon}{6} b_4 \Delta_k^2.$$

On the other hand, if $\|h_k\| > \frac{\varepsilon b_4 \Delta_k}{6\sqrt{2}b_5}$, then using (8) and Lemma 4.1, we can write

$$P_k \ge \frac{\varepsilon b_2 b_4}{12\sqrt{2}b_5} \min\{1, \frac{\varepsilon b_3 b_4}{6\sqrt{2}b_5}\}\Delta_k^2.$$

So in all cases, we have

$$P_k \ge c_2 \Delta_k^2, \tag{21}$$

where

$$c_2 = \min\left\{\frac{\varepsilon}{48\Delta_\star}\min[1, \frac{\varepsilon}{12b\Delta_\star}], \frac{\varepsilon b_4}{6}, \frac{\varepsilon b_2 b_4}{12\sqrt{2}b_5}\min[1, \frac{\varepsilon b_3 b_4}{6\sqrt{2}b_5}]\right\}.$$

Now for any iteration indexed $k \ge k_1$

$$A_k \ge \eta_1 P_k \ge \eta_1 c_2 \Delta_k^2.$$

If $k_2 \ge \max[k_0, k_1]$, then the last inequality and the fact that $\{\Phi_k\}$ is bounded below imply that

$$\infty > \sum_{k=k_2}^{\infty}(\Phi_k - \Phi_{k+1}) = \sum_{k=k_2}^{\infty} A_k \ge \sum_{k=k_2}^{\infty} \eta_1 P_k \ge \sum_{k=k_2}^{\infty} \eta_1 c_2 \Delta_k^2.$$

This implies that

$$\lim_{k\to\infty} \Delta_k = 0.$$

But using (21) and Lemma 4.6, we obtain, for all $k > k_2$

$$|\Gamma_k - 1| \le \frac{r_{k_0}}{c_2}[b_7\Delta_k + b_8\|h_k\|].$$

Because both $\|h_k\|$ and Δ_k are converging to zero, the condition $\Gamma_k \ge \eta_4$ will always be satisfied for large k. So Δ_k can not converge to zero. This is a contradiction. So the supposition was wrong and the theorem is proved. □

The above theorem shows that there exists a subsequence of iterates $\{x_{k_j}\}$ that satisfies

$$\lim_{k_j\to\infty}\left[\ \|Z_{k_j}^T \nabla f_{k_j}\| + \|h_{k_j}\|\ \right] = 0, \tag{22}$$

and

$$\lim_{k_j\to\infty} \gamma_{k_j} = 0. \tag{23}$$

If we assume that the sequence of iterates $\{x_k\}$ generated by the algorithm is bounded, then Equation (22) says that there exists a subsequence of iterates of $\{x_{k_j}\}$ converges to a point, say $x_\star$, that satisfies the first-order necessary conditions. Equation (23) means that the matrix $Z_\star^T \nabla_x^2 \ell(x_\star, \lambda_\star) Z_\star$ is positive semi-definite, if we assume the continuity of the matrix $Z(x)$ in Ω. For more details about the continuity of the matrix $Z(x)$, see Ref. 28. This result is stated in the following theorem.

Theorem 4.2. Assume (A1)-(A4). Assume further that $Z(x)$ is continuous in Ω. If the sequence of iterates generated by the algorithm is bounded, then there exists a limit point $x_\star$ that satisfies

$$\begin{aligned} Z(x_\star)^T \nabla f(x_\star) &= 0 \\ h(x_\star) &= 0, \end{aligned}$$

and the matrix $Z(x_\star)^T \nabla_x^2 \ell(x_\star, \lambda_\star) Z(x_\star)$ is positive semi-definite.

Proof. The proof follows from the above discussion. □

The above theorem implies that there exists a subsequence of the sequence of iterates generated by the algorithm that will converge to a point that satisfies the second order necessary conditions.

5 Concluding Remarks

We have presented a convergence theory to a second-order point for the algorithm that was suggested by the author in Ref. 1.

We showed that the sequence of iterates $\{x_k\}$ generated by the algorithm will not be bounded away from points that satisfy the second-order necessary conditions. In other words, we showed that there exists a subsequence of iterates that converges to a point that satisfies both the first-order and the second-order necessary conditions.

This result is similar to the second-order results obtained in Refs. 29-31 for the unconstrained optimization problem and recently in Ref. 21 for the equality constrained optimization problem.

6 References

1. EL-ALEM, M. M., *A Robust Trust Region Algorithm with a Nonmonotonic Penalty*

Parameter Scheme for Constrained Optimization, SIAM Journal on Optimization, Vol. 5, No. 2, pp. 348 - 378, 1995.

2. FIACCO, A., and MCCORMICK, G., *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley, New York, New York, 1968.

3. DENNIS, J., EL-ALEM, M., and MACIEL, M., *A Global Convergence Theory for General Trust-Region-Based Algorithms for Equality Constrained Optimization*, Technical Report 92-28, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1992.

4. WILSON, R. B., *A Simplified Algorithm for Concave Programming*, Ph D Thesis, Harvard University, Cambridge, Massachusetts, 1963.

5. TAPIA, R. A., *Quasi-Newton Methods for Equality Constrained Optimization: Equivalence of Existing Methods and a New Implementation*, Nonlinear Programming 3, Edited by O. Mangasarian, R. Meyer, and S. Robinson, Academic Press, New York, New York, pp. 125-146, 1978.

6. MIELE, A., HUANG, H. Y., and HEIDEMAN, J. C., *Sequential Gradient-Restoration Algorithm for the Minimization of Constrained Functions: Ordinary and Conjugate Gradient Versions*, Journal of Optimization Theory and Applications, Vol 4, No. 4, pp. 213-243, 1969.

7. VARDI, A., *A Trust Region Algorithm for Equality Constrained Minimization: Convergence Properties and Implementation*, SIAM Journal on Numerical Analysis, Vol. 22, pp. 575-591, 1985.

8. BYRD, R., SCHNABEL, B., and SHULTZ, G., *A Trust Region Algorithm for Nonlinearly Constrained Optimization*, SIAM Journal on Numerical Analysis, Vol 24, pp. 1152-1170, 1987.

9. CELIS, M., DENNIS, J., and TAPIA, R., *A Trust Region Strategy for Nonlinear Equality Constrained Optimization*, Numerical Optimization, Edited by P. Boggs, R. Byrd, and R. Schnabel, SIAM, Philadelphia, Pennsylvania, pp. 71-82, 1985.

10. EL-ALEM, M. M., *A Global Convergence Theory for a Class of Trust Region Algorithms for Constrained Optimization*, Ph D Thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1988.

11. EL-ALEM, M. M., *A Global Convergence Theory for the Celis-Dennis-Tapia Trust-Region Algorithm for Constrained Optimization*, SIAM Journal on Numerical Analysis, Vol. 24, pp. 266-290, 1991.

12. POWELL, M. J. D., and YUAN, Y., *A Trust Region Algorithm for Equality Constrained Optimization*, Mathematical Programming, Vol. 49, pp. 189-211, 1991.

13. EL-ALEM, M. M. and TAPIA, R. A., *Numerical Experience with a Polyhedral Norm CDT Trust-Region Algorithm*, Journal of Optimization Theory and Applications, Vol. 85, No. 3, pp. 575-592, 1995.

14. YUAN, Y., *A Dual Algorithm for Minimizing a Quadratic Function With Two Quadratic Constraints*, Technical Report DAMTP-NA3, University of Cambridge, Cambridge, England, 1988.

15. YUAN, Y., *On a Subproblem of Trust Region Algorithms For Constrained Optimization*, Mathematical Programming, Vol 47, pp. 53-63, 1990.

16. ZHANG, Y., *Computing a Celis-Dennis-Tapia Trust Region Step for Equality Constrained Optimization*, Mathematical Programming, Vol 55, pp. 109-124, 1992.

17. WILLIAMSON, K. A., *A Robust Trust Region Algorithm for Nonlinear Programming*, Ph D Thesis, Department of Mathematical Sciences, Rice University, Houston, Texas, 1990.

18. ALEXANDROV, N., *Multi-Level Algorithms for Nonlinear Equations and Equality Constrained Optimization*, Ph D Thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1993.

19. ALEXANDROV, N., and DENNIS, J. E., *Multi-Level Algorithms for Nonlinear Optimization*, Technical Report 94-24, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1994.

20. BYRD, R., and OMOJOKUN, E., *Robust Trust-Region Methods for Nonlinearly Constrained Optimization*, Paper Presented at the SIAM Conference on Optimization, Houston, Texas, 1987.

21. DENNIS, J. and VICENTE, L., *On the Convergence Theory of Trust-Region-Based Algorithms for Equality-Constrained Optimization*, Technical Report 94-36, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1994.

22. EL-ALEM, M. M., *Global Convergence Without the Assumption of Linear Independence for a Trust-Region Algorithm for Constrained Optimization*, Journal of Optimization Theory and Applications, Vol. 87, No. 3, pp. 563-577, 1995.

23. MACIEL, M. C., *A Global Convergence Theory for a General Class of Trust-Region Algorithm for Equality Constrained Optimization*, Ph D Thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1992.

24. OMOJOKUN, E., *Trust Region Strategies for Optimization with Nonlinear Equality and Inequality Constraints*, Ph D Thesis, Department of Computer Science, University of Colorado, Boulder, Colorado, 1989.

25. ZHANG, J., and ZHU, D., *Projected Quasi-Newton Algorithm with Trust Region for Constrained Optimization*, Journal of Optimization Theory and Applications, Vol 67, pp. 369-393, 1990.

26. FLETCHER, R., *Practical Methods of Optimization, Volume 1: Unconstrained Optimization*, John Wiley and Sons, New York, New York,1980.

27. MORÉ, J. J., *Recent Developments in Algorithms and Software for Trust-Region Methods*, Mathematical programming: The State of The Art, Edited by A. Bachem, M. Grotschel, and B. Korte, Springer-Verlag, New York, New York, pp. 258-287, 1983.

28. COLEMAN, T. and SORENSEN, D., *A Note on the Computation of an Orthonormal Basis for the Null Space of a Matrix*, Mathematical Programming, Vol. 29, pp. 234-242, 1984.

29. MORÉ, J. J. and SORENSEN, D., *Computing a Trust-Region Step*, SIAM Journal on Statistical and Scientific Computing Vol 4, pp. 553-572, 1983.

30. SHULTZ, G., SCHNABEL, R., and BYRD, R., *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.

31. SORENSEN, D., *Newton's Method with a Model Trust-Region Modification*, SIAM Journal on Numerical Analysis, Vol. 19, pp. 409-426, 1982.