A Robust Trust-Region Algorithm with a Non-Monotonic Penalty Parameter Scheme for Constrained Optimization

Mahmoud El-Alem

CRPC-TR92245
September 1992

Center for Research on Parallel Computation
Rice University
P.O. Box 1892
Houston, TX 77251-1892

Revised September 10, 1993

A ROBUST TRUST-REGION ALGORITHM WITH A NON-MONOTONIC PENALTY PARAMETER SCHEME FOR CONSTRAINED OPTIMIZATION[†]

MAHMOUD EL-ALEM[‡]

Abstract. An algorithm for solving the problem of minimizing a non-linear function subject to equality constraints is introduced. This algorithm is a trust-region algorithm. In computing the trial step, a projected-Hessian technique is used that converts the trust-region subproblem to a one similar to that for the unconstrained case. To force global convergence, the augmented Lagrangian is employed as a merit function.

One of the main advantages of this algorithm is the way that the penalty parameter is updated. We introduce an updating scheme that allows (for the first time to the best of our knowledge) the penalty parameter to be decreased whenever it is warranted. The behavior of this penalty parameter is studied.

A convergence theory for this algorithm is presented. It is shown that this algorithm is globally convergent and that the globalization strategy will not disrupt fast local convergence. The local rate of convergence is also discussed. This theory is sufficiently general that it holds for any algorithm that generates steps whose normal components give at least a fraction of Cauchy decrease in the quadratic model of the constraints and uses Fletcher's exact penalty function as a merit function.

Key Words : Constrained Optimization, Global Convergence, Projected Hessian, Penalty Parameter, Local Convergence, Trust Region, Equality Constrained.

AMS subject classifications. 65K05, 49D37.

1. Introduction.

In this paper, we study the following non-linear equality constrained optimization problem:

$$(\mathrm{EQ}) \equiv \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & h(x) = 0, \end{cases}$$

where $h(x) = [h_1(x), ..., h_m(x)]^T$. We assume that f and $h_i, i = 1, 2, ..., m$ are twice continuously differentiable and that ∇h has full column rank in the range of interest where $\nabla h(x) = [\nabla h_1(x), ..., \nabla h_m(x)]$.

We can obtain first and second order conditions of optimality with reference to the Lagrangian function associated with problem (EQ), namely $l(x, \lambda) = f(x) + \lambda^T h(x)$ where $\lambda \in \Re^m$ is the Lagrange multiplier vector. The first order necessary condition for a point $x_\star$ to be a stationary point of problem (EQ) is the existence of a Lagrange multiplier $\lambda_\star$ such that $(x_\star, \lambda_\star)$ is a zero of the following $(n+m) \times (n+m)$ nonlinear system of equations:

$$\begin{bmatrix} \nabla_x l(x, \lambda) \\ h(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{1.1}$$

Consider an $n \times (n-m)$ matrix $Z(x)$, with orthonormal columns that has the property $Z(x)^T \nabla h(x) = 0$. The columns of $Z(x)$ form an orthonormal basis for the null space of $\nabla h(x)^T$. The matrix $Z(x)$ can be obtained from the QR factorization of $\nabla h(x)$ as follows

$$\nabla h(x) = [\, Y(x) \quad Z(x) \,] \begin{bmatrix} R(x) \\ 0 \end{bmatrix}, \tag{1.2}$$

† Research was supported by DOE DE-FG005-86ER25017, CRPC CCR-9120008, and the REDI Foundation.

‡ Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt.

where $Y(x) \in \Re^{n \times m}$. The orthonormal columns of $Y(x)$ form a basis for the column space of $\nabla h(x)$ and $R(x)$ is an $m \times m$ nonsingular upper triangular matrix. It is easy to see that: $Y(x)^T Y(x) = I_m$, $Z(x)^T Z(x) = I_{n-m}$, and $Y(x)Y(x)^T + Z(x)Z(x)^T = I_n$.

Using this factorization, an equivalent first order necessary condition can be written in the following form:

$$\begin{bmatrix} Z(x_\star)^T \nabla f(x_\star) \\ h(x_\star) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{1.3}$$

The second order sufficiency condition for the point $x_\star$ to be a solution of problem (EQ) is the existence of a multiplier $\lambda_\star \in \Re^m$ such that the point $(x_\star, \lambda_\star)$ satisfies the first order necessary condition (1.1) and the matrix $Z(x_\star)^T \nabla_x{}^2 l(x_\star, \lambda_\star) Z(x_\star)$ is positive definite.

Throughout this paper, all the norms used are 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 $l(x_k, \lambda_k)$, and so on.

Some of the algorithms that solve problem (EQ) use Newton's method to find a zero of (1.1). This gives rise to the following $(n+m) \times (n+m)$ linear system:

$$\begin{bmatrix} \nabla_x{}^2 l_k & \nabla h_k \\ \nabla h_k^T & 0 \end{bmatrix} \begin{bmatrix} s_k \\ \Delta\lambda_k \end{bmatrix} = - \begin{bmatrix} \nabla_x l_k \\ h_k \end{bmatrix}. \tag{1.4}$$

If we pre-multiplying the first block of (1.4) by Z_k^T, we obtain the following $n \times n$ linear system:

$$\begin{bmatrix} Z_k^T \nabla_x{}^2 l_k \\ \nabla h_k^T \end{bmatrix} s_k = - \begin{bmatrix} Z_k^T \nabla f_k \\ h_k \end{bmatrix}. \tag{1.5}$$

Letting $s_k = Y_k u_k + Z_k v_k$ and using the factorization (1.2), the above system becomes

$$\begin{bmatrix} Z_k^T \nabla_x{}^2 l_k Y_k & Z_k^T \nabla_x{}^2 l_k Z_k \\ R_k^T & 0 \end{bmatrix} \begin{bmatrix} u_k \\ v_k \end{bmatrix} = - \begin{bmatrix} Z_k^T \nabla f_k \\ h_k \end{bmatrix}. \tag{1.6}$$

By solving this system of equations for u_k and v_k, we can obtain s_k. More details can be found in Gill and Murray (1974)[9] and Goodman (1985)[10].

The Lagrange multiplier λ_{k+1} is obtained using the least-squares estimate:

$$\lambda_{k+1} = \text{argmin} \|\nabla h_{k+1} \lambda + \nabla f_{k+1}\|. \tag{1.7}$$

Using (1.2), this problem is equivalent to solving $R_{k+1}\lambda_{k+1} = -Y_{k+1}^T \nabla f_{k+1}$.

We can proceed by maintaining a quasi-Newton approximation B_k to the Hessian of the Lagrangian $\nabla_x^2 l_k$ in (1.6). More details can be found in Nocedal and Overton (1985)[14]. So, the algorithm for computing the trial step s_k and the multiplier λ_{k+1} can be outlined as follows:

Algorithm 1.1

At each iteration k, do

Solve $R_k^T u_k = -h_k$, for u_k.

Solve $Z_k^T B_k Z_k v_k = -Z_k^T \nabla f_k - Z_k^T B_k Y_k u_k$, for v_k.

Set $s_k = Y_k u_k + Z_k v_k$ and $x_{k+1} = x_k + s_k$.

Find λ_{k+1} by solving $R_{k+1}\lambda_{k+1} = -Y_{k+1}^T \nabla f_{k+1}$.

End do

It is easy to see that for problem (EQ), if the exact second-order information is used, the above algorithm can be viewed as a Newton's method applied to the nonlinear system (1.1) (see Goodman (1985)[10]). Hence, it shares the advantages and the disadvantages of Newton's method. From the good side of Newton's method, it is locally q-quadratically convergent. However, from the bad side of Newton's method, it is not a globally convergent method. It is guaranteed to converge only if the starting point is close enough to the solution. This means that it may not converge at all if the starting point is far away from the solution. More details can be found in Tapia (1978)[21].

The next section deals with adding a trust-region modification to this method to force convergence to a solution from any starting point without sacrificing fast local convergence.

2. Trust-Region Globalization. The key idea of the trust-region method is to restrict the trial step to a region where you trust your model. This can be done by imposing the trust-region constraint $\|s_k\| \leq \Delta_k$, where the trust-region radius Δ_k is adjusted automatically from iteration to iteration. The intent is to reduce a merit function $\Phi(x)$ and the aim is to make the iterates $x_{k+1} = x_k + s_k$; $k = 1, 2, 3, \ldots$ acceptable points where s_k is obtained by solving some trust-region subproblems. More details about the trust-region method can be found in Dennis and Schnabel (1984)[4].

Byrd, Schnabel and Shultz (1987)[2] suggested computing the trial steps using the following technique: Set $s_k = Y_k u_k + Z_k v_k$ where Y_k and Z_k are as in (1.2). The two components u_k and v_k are computed by solving two subproblems. For computing u_k, they suggested solving the following linear system:

$$R_k^T u_k = -\alpha_k h_k,$$

where α_k is a constant that satisfies some specified conditions. The tangential component v_k is obtained by solving the following trust-region subproblem:

$$\begin{aligned} &\underset{v \in \Re^{n-m}}{\text{minimize}} \ (Z_k^T \nabla f_k + \alpha_k Z_k^T \nabla_x^2 l_k Y_k u_k)^T v_k + \tfrac{1}{2} v_k{}^T Z_k{}^T \nabla_x^2 l_k Z_k v_k \\ &\text{subject to } \|v_k\|^2 \leq \Delta_k^2 - \alpha_k^2 \|u_k\|^2. \end{aligned}$$

This approach suffers from the disadvantage that the step depends on the unknown parameter α_k and there is no clear way for choosing this parameter.

An interesting way of using this approach to compute a trial step that does not depend on the parameter α_k was suggested by Byrd and Omojokun (1989)[15]. They calculated s_k by solving two trust-region subproblems. For computing u_k, they suggested solving

$$\begin{aligned} &\underset{u \in \Re^{m}}{\text{minimize}} \ \|\nabla h_k{}^T Y_k u_k + h_k\|^2 \\ &\text{subject to } \|Y_k u_k\| \leq \tau \Delta_k, \end{aligned}$$

where $\tau \in (0, 1)$ is a constant. The tangential component is obtained by solving the following trust-region subproblem:

$$\begin{aligned} &\underset{v \in \Re^{n-m}}{\text{minimize}} \ (Z_k^T \nabla f_k + Z_k^T \nabla_x^2 l_k Y_k u_k)^T v_k + \tfrac{1}{2} v_k{}^T Z_k{}^T \nabla_x^2 l_k Z_k v_k \\ &\text{subject to } \|v_k\|^2 \leq \Delta_k^2 - \|Y_k u_k\|^2. \end{aligned}$$

To force global convergence. Byrd, Schnabel and Shultz (1987)[2] and Byrd and Omojokun (1989)[15] employed a non-differentiable merit function. This type of merit function suffers from the Maratos effect which may disrupt fast local convergence. See Maratos (1978)[12].

To avoid the Maratos effect, they suggested adding to the step what is called the second-order correction, and is a step of the form $w_k = -R_k^{-T} h_{k+}$ where k_+ is an intermediate point. See also Coleman and Conn (1982)[3], Fletcher (1982)[7], and Mayne and Polak (1982)[13]. However, this approach adds extra expense to the step calculation since it requires an extra constraint evaluation to compute a trial step.

In this paper, we use an inexpensive way to compute the trial steps. We employ, as a merit function, a differentiable penalty function. We will use, Fletcher's exact penalty function:

$$\Phi(x, \lambda; r) = f(x) + \lambda(x)^T h(x) + r\|h(x)\|^2, \tag{2.1}$$

where λ is the least-squares estimate of the multiplier and r is the penalty parameter. We introduce a new non-monotonic penalty parameter scheme. This penalty parameter is very cheap to calculate.

We present a convergence theory for this algorithm. Our global convergence theory is so general that it covers the algorithm of Byrd, Schnabel and Shultz (1987)[2] and the algorithm of Byrd and Omojokun (1989)[15] provided that (2.1) is used as a merit function and Scheme 3.4 (see Section 3.3) is used for updating the penalty parameter.

The remainder of this paper is organized as follows: In Section 3, we describe in detail the trust-region subproblems that will be considered and the way of computing the trial steps. A scheme for updating the radius of the trust region is presented together with a discussion about the criteria for accepting or rejecting the trial steps. Our new scheme for updating the penalty parameter will be presented in this section as well as the algorithm. In Section 4, we state the global assumptions under which we prove global convergence. In Section 5, we present our global convergence theory. We start with presenting some needed intermediate results together with some lemmas that analyze the behavior of the penalty parameter. We end this section by presenting the main global convergence results of our algorithm. In Section 6, we present the local convergence analysis. Section 7 contains concluding remarks.

3. The Trust-Region Algorithm. The algorithm has four main ingredients. The first one is computing the trial step. It is discussed in Section 3.1. The second one is testing the step and updating the trust-region radius and is discussed in Section 3.2. The third one is updating the penalty parameter and is discussed in Section 3.3. The fourth ingredient of our algorithm is how to update the matrix B_k. This will be discussed at the end of Section 3.3.

3.1. Computing The Trial Steps. In our trust-region algorithm, at each iteration, two model subproblems are solved to obtain a trial step s_k. Our way of computing the trial step is similar to that of Byrd, Schnabel and Schultz (1987)[2] with a simpler way of determining the parameter α_k (see Section 2). We start by solving for u_k the following linear system of equations

$$R_k^T u_k = -h_k, \tag{3.1}$$

then we control the size of this step by solving for α_k the following one dimensional minimization problem:

$$\begin{array}{ll} \underset{\alpha_k \in \Re}{\text{minimize}} & \|h_k + \alpha_k \nabla h_k^T Y_k u_k\| \\ \text{subject to} & \alpha_k \|u_k\| \leq \tau \Delta_k, \end{array}$$

where $\tau \in (0, 1]$ is a fixed constant. This is equivalent to setting

$$\alpha_k = \begin{cases} 1 & \text{if } \|u_k\| \leq \tau\Delta_k \\ \frac{\tau\Delta_k}{\|u_k\|} & \text{if } \|u_k\| > \tau\Delta_k. \end{cases} \tag{3.2}$$

See Zhang and Zhu (1990)[24].

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

$$\underset{v \in \Re^{n-m}}{\text{minimize}} \ (Z_k^T \nabla f_k + \alpha_k Z_k^T B_k Y_k u_k)^T v_k + \tfrac{1}{2} v_k{}^T Z_k^T B_k Z_k v_k \tag{3.3}$$

$$\text{subject to } \|v_k\| \leq \Delta_k, \tag{3.4}$$

where B_k is the Hessian of the Lagrangian $\nabla_x^2 l_k$ or an approximation to it.

The trial step will then have the form $s_k = \alpha_k Y_k u_k + Z_k v_k$. This can be outlined in the following scheme:

Scheme 3.1 Computing the trial steps
Given $0 < \tau \leq 1$.
At each iteration k, do
  Solve (3.1) for u_k, then find α_k using (3.2).
  Solve (3.3) and (3.4) for v_k.
  Set $s_k = \alpha_k Y_k u_k + Z_k v_k$ and set $x_{k+1} = x_k + s_k$.
  Find λ_{k+1} by solving $R_{k+1}\lambda_{k+1} = -Y_{k+1}^T \nabla f_{k+1}$.
End do.

Byrd and Omojokun's way of computing the normal component $s_k^n = Y_k u_k$ is more expensive since, to compute u_k, it requires solving a trust-region subproblem at each trial step. Our way requires computing u_k only once per acceptable step. Namely, when the algorithm moves to a new point after finding an acceptable step. To compute u_k, we solve (3.1) which is an upper triangular linear system. Y_k and R_k are obtained with no extra cost, since they are obtained from the QR factorization that was performed to compute the multiplier of the last acceptable step.

3.2. Testing the Step and Updating the Trust-Region Radius. Let $x_{k+1} = x_k + s_k$ where s_k is the step computed by the algorithm and λ_{k+1} be the corresponding Lagrange multiplier, we test whether the point (x_{k+1}, λ_{k+1}) is making a progress towards a solution (x_*, λ_*). In order to do this we use, as a merit function, Fletcher's exact penalty function (2.1). We test (x_{k+1}, λ_{k+1}) to determine whether it makes an improvement in the merit function.

We define the actual reduction in the merit function in moving from (x_k, λ_k) to (x_{k+1}, λ_{k+1}) to be

$$Ared_k = \Phi(x_k, \lambda_k; r_k) - \Phi(x_{k+1}, \lambda_{k+1}; r_k),$$

which can be written as

$$Ared_k = l(x_k, \lambda_k) - l(x_{k+1}, \lambda_k) - (\lambda_{k+1} - \lambda_k)^T h_{k+1} + r_k[\|h_k\|^2 - \|h_{k+1}\|^2].$$

The calculation of the step s_k is based on a quadratic approximation of the Lagrangian function and a linear approximation to the constraints. Using these approximations in a straightforward manner (see Maciel (1992)[11]), the predicted reduction will have the form

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

This form of $Pred_k$ has been used by Maciel (1992)[11]. An undesirable property of using the above expression is that $Pred_k$ depends on $\nabla \lambda_k$ which requires the evaluation of the Hessians of the objective function and the constraints. In order to avoid these calculations, the following form of predicted reduction can be used:

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

This expression for $Pred_k$ has been used by El-Alem (1988)[5] and (1991)[6]. Our definition of the predicted reduction has the form:

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

The above expression for $Pred_k$ was also used by Powell and Yuan (1991)[19]. They pointed out that the presence of the terms $\frac{1}{2} s_k^T B_k Z_k v_k$ instead of $\frac{1}{2} s_k^T B_k s_k$ and the term $h_k + \frac{1}{2} \nabla h_k^T s_k$ instead of $h_k + \nabla h_k^T s_k$ will allow for a Q-superlinear rate of convergence. See Section 6 for more details about these terms and how they will allow for Q-superlinear rate of convergence.

The normal predicted decrease and the tangential predicted decrease are also considered. They are denoted by $Npred_k$ and $Tpred_k$ respectively. The $Npred_k$ is the decrease at the $k^{\underline{th}}$ iteration in the linearized model of the constraints by the step $s_k^n = \alpha_k Y_k u_k$ and is defined by:

$$Npred_k = ||h_k||^2 - ||h_k + \alpha_k \nabla h_k^T Y_k u_k||^2.$$

It predicts the actual reduction in the constraints obtained by the normal component s_k^n.

The $Tpred_k$ is the decrease at the $k^{\underline{th}}$ iteration in the quadratic model of the Lagrangian by the step $s_k^t = Z_k v_k$. It predicts the actual reduction in the Lagrangian function obtained by the tangential component s_k^t. It is defined by:

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

The trust-region algorithm should produce steps that result in decrease in the merit function Φ. To guarantee this, the predicted reduction has to be greater than zero and the actual reduction has to be greater than some fraction of the predicted reduction. Therefore, at each iteration, the penalty parameter r_k is chosen such that $Pred_k > 0$ and the step is accepted if $\frac{Ared_k}{Pred_k} \geq \eta_1 > 0$ where $\eta_1 \in (0,1)$ is a small fixed constant. We reject the step if $\frac{Ared_k}{Pred_k} < \eta_1$. 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 $0 < a_1 \leq a_2 < \frac{1}{\sqrt{1+\tau^2}}$ and then go back and compute another trial step with new value of the trust-region radius.

If the step is accepted, then the trust-region radius is updated by comparing the value of $Ared_k$ with $Pred_k$. Namely, if $\eta_1 \leq \frac{Ared_k}{Pred_k} < \eta_2$ where $\eta_2 \in (\eta_1, 1)$, then the radius of the trust region is updated by the rule: $\Delta_{k+1} = \min[\Delta_k, a_3\|s_k\|]$ where $a_3 > \frac{1}{\sqrt{1+\tau^2}}$. However, if $\frac{Ared_k}{Pred_k} \geq \eta_2$, then we increase the radius of the trust region by setting $\Delta_{k+1} = \min[\Delta_*, \max(\Delta_k, a_3\|s_k\|)]$, where Δ_* is a positive constant. This can be summarized in the following scheme.

Scheme 3.2 Testing the Step and Updating the Trust-Region Radius
Given $0 < a_1 \leq a_2 < \frac{1}{\sqrt{1+\tau^2}} < a_3$, $0 < \eta_1 < \eta_2 < 1$ and $\Delta_* \geq \Delta_1 > 0$.
At each iteration k, do
  If $\frac{Ared_k}{Pred_k} < \eta_1$,
    then set $\Delta_k \in [a_1\|s_k\|, a_2\|s_k\|]$.
    goto Scheme 3.1 to find another trial step.
    Else, if $\eta_1 \leq \frac{Ared_k}{Pred_k} < \eta_2$
      then set $x_{k+1} = x_k + s_k$,
        $\Delta_{k+1} = \min[\Delta_k, a_3\|s_k\|]$.
      Else, set $x_{k+1} = x_k + s_k$,
        $\Delta_{k+1} = \min[\Delta_*, \max(\Delta_k, a_3\|s_k\|)]$.
    End if
  End if.
End do

The index k is increased only if the step is accepted. We use the notation k^j to denote the $\underline{jth}$ unacceptable trial step of iteration k.

It is worth noting that, under suitable assumptions, after a finite number of trial steps an acceptable step will be found, i.e. the condition $\frac{Ared_{k^j}}{Pred_{k^j}} \geq \eta_1$ will be satisfied for some j. See Theorem 5.7.

3.3. Updating the Penalty Parameter. Now, we describe our strategy for updating the penalty parameter r. The author in (1988)[5] and (1991)[6] has suggested a scheme for updating the penalty parameter. The idea behind that scheme was to keep the penalty parameter as small as possible subject to satisfying conditions needed to prove global convergence. One of these conditions was that the sequence $\{r_k\}$ of penalty parameter must be nondecreasing. If that scheme were implemented in our problem, the scheme would be as follows:

Scheme 3.3 El-Alem (1988)
Given a constant $\rho > 0$ and $r_o = 1$:
At each iteration k, do
  Set $r_k = r_{k-1}$.
  If $Pred_k < \frac{r_{k-1}}{2}[\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2]$,
    then set

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

  End if
End do.

Even though when this scheme was implemented, good performance was reported, (see Williamson (1990)[23]), this way of updating the penalty parameter has the disadvantage of producing a nondecreasing sequence of penalty parameters. This means if at one iteration the value of the penalty parameter is large, all the subsequent penalty parameters will remain at least as large as this one. Hence, the problem of obtaining feasibility has more weight than the problem of obtaining optimality. As a consequence we may progress too fast toward nonlinear feasibility at the expense of optimality. On the other hand, numerical experiments have suggested that efficient performance of the algorithm is linked to keeping the penalty parameter as small as possible (see Gill, Murray, Saunders, and Wright (1986)[8]). We propose a scheme that allows (for the first time to the best of our knowledge) the penalty parameter to be decreased whenever it is warranted.

Our convergence theory requires that the predicted reduction in the merit function at each iteration be at least as much as a fraction of Cauchy decrease in the 2-norm of the residual of the linearized constraints. (For more detail about the fraction of Cauchy decrease condition see, for example, Powell (1975)[16]). Hence, we will ask for this condition to be satisfied at each iteration.

Our convergence theory allows the sequence $\{r_k\}$ to be non-monotonic, provided that it is controlled by a sequence $\{\underline{\rho}_k\}$, which we introduce below, in the sense that for all k, $\underline{\rho}_{k-1} \leq r_k$.

So, our strategy will be, at each iteration k, pick a number $r_k \geq \underline{\rho}_{k-1}$. Then test for inequality (3.7) (see below) to be satisfied or update the penalty parameter using (3.6) (see below) which enforces (3.7). This scheme can be stated as follows:

Scheme 3.4 Updating the Penalty Parameter

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

Set $r_o = r_{-1} = \ldots\ldots = r_{-N+1} = 1$

At each iteration k, do

Find $\underline{\rho}_{k-1} = \min\{r_{k-1}, r_{k-2}, \ldots\ldots\ldots, r_{k-N}\}$,

$\overline{\rho}_{k-1} = \max\{r_{k-1}, r_{k-2}, \ldots\ldots\ldots, r_{k-N}\}$.

Set

$$\rho_{k-1} = \min\{\ \underline{\rho}_{k-1} + \rho\ ,\ \overline{\rho}_{k-1}\ \}. \tag{3.5}$$

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

If

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

then set

$$r_k = 2\{\frac{\nabla l_k^T s_k + \frac{1}{2} s_k^T B_k Z_k v_k + (\lambda_{k+1} - \lambda_k)^T [h_k + \frac{1}{2}\nabla h_k^T s_k]}{\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2}\} + \rho\ . \tag{3.6}$$

End if

End do.

The following are noteworthy;

1) The way of updating the penalty parameter ensures a predicted decrease in the merit function given by:

$$Pred_k \geq \frac{r_k}{2}[\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2]. \tag{3.7}$$

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:

$$Pred_k \geq \frac{r_k}{2} Npred_k \, . \tag{3.8}$$

2) If $N = 1$, then Scheme 3.4 will coincide with Scheme 3.3.
3) In the implementation, if we take N equal to the maximum number of iterations allowed, then we will have a scheme for updating the penalty parameter that has no requirement on r_k except that it satisfies inequality (3.7).
4) The sequence $\{\underline{\rho}_k\}$ is a monotonically non-decreasing sequence. (See Section 5.2 for a proof). But the sequence $\{\overline{\rho}_k\}$ is a non-monotonic sequence and only satisfies, for all k, $\underline{\rho}_k \leq r_k \leq \overline{\rho}_k$. This inequality shows that even though the sequence $\{r_k\}$ is a non-monotonic sequence, it is controlled by the two sequences $\{\underline{\rho}_k\}$, $\{\overline{\rho}_k\}$.
5) If at any iteration k we have $\frac{Ared_k}{Pred_k} < \eta_1$, then we reject the trial step and also reject the value of the penalty parameter. *i. e.* the only change in the problem by an unacceptable trial step is a decrease in the radius of the trust region. Observe that the unacceptable trial iterations are not included in the definition of $\underline{\rho}_k$ and $\overline{\rho}_k$.

Finally, we discuss our strategy for updating the matrix B_k. If the exact Hessian is used, then at each iteration k we compute $\nabla_x^2 l_k = \nabla^2 f_k + \nabla^2 h_k \lambda_k$. Otherwise, update B_k by some updating formula that satisfies the Global Assumption 5 (see Section 4) if we are interested in obtaining only global convergence regardless of the rate of convergence, or, satisfies the Global Assumption 5 and the Local Assumption C (see Section 6.3) if we are interested in obtaining global convergence with a fast local rate of convergence.

3.4. Statement of The Algorithm. The following is an outline of the algorithm.

Choose $x_0 \in \Re^n$, $\varepsilon > 0$, and $B_0 \in \Re^{n\times n}$.
Set $k = 0$.
At each iteration k, do
  If $\|Z_k^T \nabla f_k\| + \|h_k\| < \varepsilon$, stop.
  Compute s_k, λ_{k+1} according to Scheme 3.1.
  Update the penalty parameter according to Scheme 3.4.
  Test the step and update Δ_k according to Scheme 3.2.
  Update B_k (see Section 3.3).
  Set $k := k + 1$.
End do.

4. The Global Assumptions. In this section we state the assumptions under which we prove global convergence.

Let the sequence of iterates generated by the algorithm be $\{x_k\}$, for such a sequence we assume,

1. For all k, x_k and $x_k + s_k \in \Omega$ where $\Omega \in \Re^n$ is a convex set.
2. f and $h_i \in C^2(\Omega)$ $i = 1, ..., m$.
3. $\nabla h(x)$ has full column rank for all $x \in \Omega$.
4. $f(x), h(x), \nabla h(x), \nabla f(x), \nabla^2 f(x)$, $R(x)^{-1}$ and each $\nabla^2 h_i(x)$, for $i = 1, ..., m$ are all uniformly bounded in norm in Ω.
5. The sequence of matrices $\{B_k\}, k = 1, 2, ...$ is bounded.

An immediate consequence of the global assumptions is that the matrices B_k, $Z_k^T B_k Z_k$, and $Z_k^T B_k Y_k$ have a uniform upper bound. *i. e.* there exists a constant $b > 0$, such that, for all k,

$$\|B_k\| \leq b, \quad \|Z_k^T B_k Z_k\| \leq b, \text{ and } \|Z_k^T B_k Y_k\| \leq b. \tag{4.1}$$

Another immediate consequence of these assumptions is the existence of constants $b_0 > 0$, $b_1 > 0$, $b_2 > 0$, and $b_3 > 0$ such that, for all k,

$$\|u_k\| \leq b_0\|h_k\|, \tag{4.2}$$

$$\|\lambda_{k+1} - \lambda_k\| \leq b_1\|s_k\|, \tag{4.3}$$

$$\|h_{k+1} - h_k\| \leq b_2\|s_k\|, \tag{4.4}$$

and

$$\|\nabla h_k\| \leq b_3. \tag{4.5}$$

If Ω were a compact set, Assumption 4 would follow from the continuity assumption.

The same assumptions as our global assumptions are used by Byrd, Schnabel, and Shultz (1987)[2], El-Alem (1988)[5] and (1991)[6] and Powell and Yuan (1991)[19].

5. Global Convergence Analysis. In this section we present our global convergence theory. In Section 5.1, we prove some intermediate lemmas needed for proving global convergence. The behavior of the penalty parameter is discussed in Section 5.2. Section 5.3 is devoted to proving our main global convergence results.

5.1. Sufficient Decrease in the Model. All the results in this section deal with the decrease in the model obtained by the trial steps and their tangential and normal components.

The following lemma shows how accurate our definition of predicted reduction in the merit function is as an approximation to the actual reduction. It says that, if the penalty parameter is bounded, it is accurate to within the square of the length of the trial steps.

LEMMA 5.1. *Let the global assumptions hold. Then, for any x_k, $x_k + s_k \in \Omega$, we have*

$$|Ared_k - Pred_k| \leq b_4 r_k \|s_k\|^2, \tag{5.1}$$

where b_4 is a positive constant independent of k.

Proof. The proof is similar to the proof of Corollary 6.4 of El-Alem (1991)[6]. Note that in the proof inequalities (4.1), (4.2), and the fact that $\|Z_k v_k\| \leq \|s_k\|$ are used. □

The following lemma shows that, at any iteration k, the normal predicted reduction $Npred_k$ is at least equal to the decrease in the 2-norm of the linearized constraints obtained by the Cauchy step. *i. e.* it satisfies the fraction of Cauchy decrease condition.

LEMMA 5.2. *At any iteration k, we have*

$$Npred_k \geq \|h_k\| \min[\|h_k\|, \frac{\tau\Delta_k}{b_0}] \,, \tag{5.2}$$

where b_0 is as in (4.2).

Proof. From the definition of $Npred_k$, we need to show that

$$\|h_k\|^2 - \|h_k + \alpha_k \nabla h_k^T Y_k u_k\|^2 \geq \|h_k\| \min[\|h_k\|, \frac{\tau\Delta_k}{b_0}] .$$

When $\|h_k\| = 0$ the above inequality is true a fortiori. Let $\|h_k\| > 0$ and consider

$$\|h_k\|^2 - \|h_k + \alpha_k \nabla h_k^T Y_k u_k\|^2 = \|h_k\|^2 - \|h_k + \alpha_k R_k^T u_k\|^2 = [1-(1-\alpha_k)^2]\|h_k\|^2.$$

We consider two cases:
First, when $\|u_k\| \leq \tau\Delta_k$. In this case $\alpha_k = 1$ and we obtain

$$\|h_k\|^2 - \|h_k + \alpha_k \nabla h_k^T Y_k u_k\|^2 = \|h_k\|^2.$$

Second, when $\|u_k\| > \tau\Delta_k$, then $\alpha_k = \frac{\tau\Delta_k}{\|u_k\|}$ and using $0 < \alpha_k \leq 1$, we get

$$\|h_k\|^2 - \|h_k + \alpha_k \nabla h_k^T Y_k u_k\|^2 \geq \alpha_k \|h_k\|^2 = \frac{\tau\Delta_k}{\|u_k\|}\|h_k\|^2.$$

Using (4.2), we obtain

$$\|h_k\|^2 - \|h_k + \alpha_k \nabla h_k^T Y_k u_k\|^2 \geq \frac{\tau\Delta_k}{b_0}\|h_k\|.$$

Now, if we combine the two cases, we get the desired result. □

If we substitute (5.2) in (3.8), we obtain

$$Pred_k \geq \frac{r_k}{2} \|h_k\| \min[\|h_k\|, \frac{\tau\Delta_k}{b_0}] . \tag{5.3}$$

From the last lemma, using (1.2), we can write

$$\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2 \geq \|h_k\| \min [\|h_k\|, \frac{\tau\Delta_k}{b_0}]. \tag{5.4}$$

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

LEMMA 5.3. *For all k, the tangential predicted reduction satisfies:*

$$Tpred_k \geq \frac{1}{4}\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \min[\Delta_k, \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{2b}] . \tag{5.5}$$

where s_k^n is the normal component of the step s_k and b is as in (4.1).

Proof. We first prove that

$$-(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k \geq \frac{1}{2}\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \min[\Delta_k, \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{2\|Z_k^T B_k Z_k\|}] .$$

When $\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| = 0$ the above inequality is valid a fortiori.
Let $\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| > 0$. If $\|v_k\| < \Delta_k$, then from the way of computing v_k, we have $Z_k^T B_k Z_k v_k + Z_k^T \nabla f_k + Z_k^T B_k s_k^n = 0$, and we can write

$$\begin{aligned}(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T \quad v_k &= -v_k^T Z_k^T B_k Z_k v_k \\ &= -(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T (Z_k^T B_k Z_k)^+ (Z_k^T \nabla f_k + Z_k^T B_k s_k^n),\end{aligned}$$

where $(Z_k^T B_k Z_k)^+$ is the generalized inverse of $Z_k^T B_k Z_k$. We have

$$(5.6) \qquad (Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k \leq -\frac{1}{\|Z_k^T B_k Z_k\|}\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|^2.$$

On the other hand, if $\|v_k\| = \Delta_k$, then from the way of computing v_k, there exists a constant $\mu_k \geq 0$ such that

$$(5.7) \qquad (Z_k^T B_k Z_k + \mu_k I)v_k + Z_k^T \nabla f_k + Z_k^T B_k s_k^n = 0.$$

This equation implies that

$$\begin{aligned}(Z_k^T \nabla f_k \quad + \quad & Z_k^T B_k s_k^n)^T v_k = -v_k^T (Z_k^T B_k Z_k + \mu_k I)v_k \\ = \quad & -(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T (Z_k^T B_k Z_k + \mu_k I)^+ (Z_k^T \nabla f_k + Z_k^T B_k s_k^n),\end{aligned}$$

which implies that

$$(5.8) \qquad (Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k \leq -\frac{1}{\overline{\lambda}(Z_k^T B_k Z_k) + \mu_k}\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|^2,$$

where $\overline{\lambda}(Z_k^T B_k Z_k)$ is the largest eigenvalue of $Z_k^T B_k Z_k$. On the other hand, from (5.7), we have

$$[\underline{\lambda}(Z_k^T B_k Z_k) + \mu_k]\|v_k\| \leq \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|,$$

where $\underline{\lambda}(Z_k^T B_k Z_k)$ is the smallest eigenvalue of $Z_k^T B_k Z_k$. The above inequality implies that

$$(5.9) \qquad \mu_k \leq \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{\Delta_k} - \underline{\lambda}(Z_k^T B_k Z_k).$$

By substituting (5.9) in (5.8), we obtain

$$(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k \leq -\frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|^2 \Delta_k}{[\overline{\lambda}(Z_k^T B_k Z_k) - \underline{\lambda}(Z_k^T B_k Z_k)]\Delta_k + \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}.$$

Now, using the fact that $\overline{\lambda}(Z_k^T B_k Z_k) - \underline{\lambda}(Z_k^T B_k Z_k) \leq 2\|Z_k^T B_k Z_k\|$, the above inequality becomes

$$(5.10) \qquad (Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k \leq -\frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|^2 \Delta_k}{2\|Z_k^T B_k Z_k\|\Delta_k + \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}.$$

So, from (5.6) and (5.10), we conclude that in both cases we can write

$$-(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k \geq \frac{1}{2}\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \min[\Delta_k, \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{2\|Z_k^T B_k Z_k\|}].$$

The rest of the proof follows directly from the definition of $Tpred_k$, the fact that $(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k + v_k^T Z_k^T B_k Z_k v_k \leq 0$, and (4.1). □

LEMMA 5.4. *Let s_k be the step generated by the algorithm at the k^{th} iteration, then*

$$Pred_k \geq Tpred_k - b_5\|s_k\|\|h_k\| + \frac{r_k}{2}Npred_k,$$

where b_5 is a positive constant independent of k.

Proof. From the definition of $Pred_k$, we have

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

This can be written as

$$\begin{aligned} Pred_k &= -(Z_k^T\nabla f_k + Z_k^T B_k s_k^n)^T v_k - \frac{1}{2}v_k^T Z_k^T B_k Z_k v_k + \frac{1}{2}\alpha_k u_k^T Y_k^T B_k Z_k v_k \\ &\quad -(\lambda_{k+1}-\lambda_k)^T[h_k + \frac{1}{2}\nabla h_k^T s_k] + r_k[\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2] \\ &\geq Tpred_k - \|\lambda_{k+1}-\lambda_k\|\|h_k + \frac{1}{2}\nabla h_k^T s_k\| - \|Y_k^T B_k Z_k\|\|v_k\|\|u_k\| \\ &\quad + \frac{r_k}{2}Npred_k. \end{aligned}$$

Using (4.1), (4.2), (4.3), the fact that $\|h_k + \frac{1}{2}\nabla h^T s_k\| \leq \|h_k\|$, and $\|v_k\| \leq \|s_k\|$, The remainder of the proof follows immediately. □

The following lemma proves that if $\|h_k\|$ is small enough, then the penalty parameter will not be updated using (3.6). *i. e.* inequality (3.7) will hold for $r_k = \rho_{k-1}$. (See Scheme 3.4).

LEMMA 5.5. *Let k be the index of an iteration at which the algorithm does not terminate. If $\|h_k\| \leq c_1\Delta_k$ where c_1 is a small constant that satisfies:*

$$c_1 \leq \min\{\frac{\varepsilon}{3\Delta_*}, \frac{\varepsilon}{3bb_0\Delta_*}, \frac{\varepsilon}{24\sqrt{2}b_5\Delta_*}\min(1, \frac{\varepsilon}{6b\Delta_*})\}, \tag{5.11}$$

then

$$Pred_k \geq \frac{1}{2}Tpred_k + \frac{r_k}{2}Npred_k. \tag{5.12}$$

Proof. From Lemma 5.4 and Lemma 5.3, we can write

$$\begin{aligned} Pred_k &\geq \frac{1}{2}Tpred_k + \frac{1}{8}\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|\min[\Delta_k, \frac{\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|}{2b}] \\ &\quad -b_5\|s_k\|\|h_k\| + \frac{r_k}{2}Npred_k. \end{aligned} \tag{5.13}$$

Since $c_1 \leq \frac{\varepsilon}{3\Delta_*}$, then $\|h_k\| \leq \frac{\varepsilon}{3}$ and because the algorithm does not terminate, $\|Z_k^T\nabla f_k\| > \frac{2\varepsilon}{3}$, and we obtain

$$\begin{aligned} \|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\| &\geq \|Z_k^T\nabla f_k\| - \|Z_k^T B_k Y_k\|\|u_k\|, \\ &\geq \frac{2\varepsilon}{3} - bb_0\|h_k\| \geq \frac{2\varepsilon}{3} - \frac{\varepsilon}{3} = \frac{\varepsilon}{3}. \end{aligned}$$

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

$$\begin{aligned} Pred_k &\geq \frac{1}{2}Tpred_k + \frac{\varepsilon\Delta_k}{24}\min[1, \frac{\varepsilon}{6b\Delta_*}] - \sqrt{2}c_1 b_5\Delta_k^2 + \frac{r_k}{2}Npred_k, \\ &\geq \frac{1}{2}Tpred_k + \{\frac{\varepsilon}{24}\min[1, \frac{\varepsilon}{6b\Delta_*}] - \sqrt{2}c_1 b_5\Delta_*\}\Delta_k + \frac{r_k}{2}Npred_k. \end{aligned}$$

From (5.11), the quantity $\{\frac{\epsilon}{24}\min[1,\frac{\epsilon}{6b\Delta_*}] - \sqrt{2}c_1 b_5 \Delta_*\}$ is positive. Hence,

$$Pred_k \geq \frac{1}{2}Tpred_k + \frac{r_k}{2}Npred_k,$$

which is the desired result. □

From the proof of the above lemma, we see that the fourth term in (5.13) did not enter into the calculation. This implies that if we set $r_k = \rho_{k-1}$ (see Scheme 3.4) inequality (5.12) remains valid. So, when $\|h_k\| \leq c_1\Delta_k$, the algorithm will not update the penalty parameter using (3.6). In other words, inequality (3.7) will always be satisfied.

LEMMA 5.6. *If the algorithm does not terminate, then any iteration at which $\|h_k\| \leq c_1\Delta_k$, satisfies*

$$Pred_k \geq c_2\Delta_k \tag{5.14}$$

where c_1 is given by (5.11) and c_2 is a positive constant independent of k.

Proof. When $\|h_k\| \leq c_1\Delta_k$, where c_1 is given by (5.11), then from Lemma 5.3 and Lemma 5.5, we have

$$Pred_k \geq \frac{1}{2}Tpred_k \geq \frac{1}{8}\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|\min[\Delta_k, \frac{\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|}{2b}].$$

But, since $c_1 < \frac{\epsilon}{3\Delta_*}$, then $\|h_k\| < \frac{\epsilon}{3}$ and because the algorithm does not terminate, we have $\|Z_k^T\nabla f_k\| \geq \frac{2\epsilon}{3}$. Thus, as in Lemma 5.5, we conclude that $\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\| \geq \frac{\epsilon}{3}$. Hence,

$$Pred_k \geq \frac{\epsilon}{24}\min[1,\frac{\epsilon}{6b\Delta_*}]\Delta_k.$$

The result now follows if we set $c_2 = \frac{\epsilon}{24}\min[1,\frac{\epsilon}{6b\Delta_*}]$. □

The following theorem shows that the algorithm is well defined in the sense that it will never loop ad infinitum without finding an acceptable step.

THEOREM 5.7. *Let the global assumptions hold. At any iteration k at which the penalty parameter r_k is bounded, either the termination condition of the algorithm will be met or an acceptable step will be found.*

Proof. In the proof of this lemma we use the notation k^j to mean the $j^{\underline{th}}$ unacceptable trial step of iteration k.

If the termination condition of the algorithm is satisfied, then there is nothing to prove. Assume that the point (x_k, λ_k) does not satisfy the termination condition of the algorithm.

Suppose that at iteration k the algorithm loops infinitely without finding an acceptable step. Hence all the trial steps are rejected and we obtain, for all j

$$(1-\eta_1) < \left|\frac{Ared_{k^j}}{Pred_{k^j}} - 1\right|. \tag{5.15}$$

First, assume that $\|h_k\| = 0$. Therefore, for all j we have $\|h_{k^j}\| \leq c_1\Delta_{k^j}$, where c_1 is as in (5.11). In this case the penalty parameter remains the same. So, we have $r_{k^j} = r_k$ is bounded for all j.
On the other hand, from Lemmas 5.1 and 5.6, for any j such that $\Delta_{k^j} > 0$, we have

$$\left|\frac{Ared_{k^j}}{Pred_{k^j}} - 1\right| = \frac{|Ared_{k^j} - Pred_{k^j}|}{Pred_{k^j}} \leq \frac{b_4 r_k}{c_2}\Delta_{k^j}.$$

As j goes to infinity, Δ_{kj} goes to zero and we get a contradiction with (5.15). So j can not go to infinity. But this contradicts the supposition that the algorithm loops infinitely without finding an acceptable step and means that, after finitely many rejected trial steps, an acceptable one will be found.

Now assume that $\|h_k\| > 0$. From (3.7), (5.2), and Lemma 5.1, we can write

$$\left|\frac{Ared_{kj}}{Pred_{kj}} - 1\right| = \frac{|Ared_{kj} - Pred_{kj}|}{Pred_{kj}} \leq \frac{2b_4\Delta_{kj}^2}{\|h_{kj}\| \min\{\|h_{kj}\|, \frac{\tau\Delta_{kj}}{b_0}\}}.$$

Here $\|h_{kj}\| = \|h_k\| > 0$ is fixed. Therefore, for sufficiently large j, we have,

$$\min\{\|h_{kj}\|, \frac{\tau\Delta_{kj}}{b_0}\} = \frac{\tau\Delta_{kj}}{b_0}.$$

Hence,

$$\left|\frac{Ared_{kj}}{Pred_{kj}} - 1\right| \leq \frac{2b_4 b_0}{\tau\|h_k\|}\Delta_{kj}.$$

As j goes to infinity, Δ_{kj} goes to zero and we get a contradiction with (5.15). So j can not go to infinity. Again this contradicts the supposition. Hence the supposition is wrong and the theorem is proved. □

Under the assumption that the algorithm does not terminate, the above theorem is true at any iteration k at which r_k is bounded. In the following section we prove that the penalty parameter is bounded for all k. This will imply that Theorem 5.7 is true for all k.

5.2. The Behavior of the Penalty Parameter . In our analysis of the penalty parameter, the sequences $\{\underline{\rho}_k\}$ and $\{\overline{\rho}_k\}$ are used. For their definitions see Scheme 3.4.

Our goal is to prove that there exists a constant $r_\star$ and an integer $\overline{k}$ such that $r_k = r_\star$ for all $k \geq \overline{k}$. To this end, we will prove the following. First we will prove that $\{\overline{\rho}_k\}$ is bounded. This of course will imply that $\{r_k\}$ and $\{\underline{\rho}_k\}$ are bounded. Second we will show that $\{\underline{\rho}_k\}$ is a non-decreasing sequence. We will also discuss the amount of increase in the sequence $\{\underline{\rho}_k\}$. Finally we will show that the sequences $\{\underline{\rho}_k\}$, $\{r_k\}$, $\{\overline{\rho}_k\}$ will attain the same value after finitely many iterations. We start with the following lemma which we will use to conclude that r_k is bounded.

LEMMA 5.8. *Under the global assumptions, the sequence $\{\overline{\rho}_k\}$ is bounded.*

Proof. If the algorithm terminates, $\{\overline{\rho}_k\}$ is finite and trivially bounded. So, consider the case when the algorithm does not terminate. The proof is by contradiction. Suppose that the sequence $\{\overline{\rho}_k\}$ is not bounded. Then there exists an infinite sequence of indices $\{k_i\}$, such that

$$\overline{\rho}_k > \max\{\frac{\sqrt{2}b_o(2b_1 + b_0 b + 2\rho b_3)}{\min(\tau, c_1 b_0)}, \overline{\rho}_1\}, \tag{5.16}$$

for all $k \in \{k_i\}$. Suppose that m is the first index such that (5.16) holds. It is clear, using inequality (5.16), that $m \geq 2$.

The only possibility that $\overline{\rho}_m > \overline{\rho}_{m-1}$ is when $r_m > \overline{\rho}_{m-1}$ and this can only happen when r_m is updated by (3.6). This implies that

$$\begin{aligned} r_m[\|h_m\|^2 - \|h_m + \nabla h_m^T s_m\|^2] &= 2(Z_m^T \nabla f_m)^T v_m + s_m^T B_m Z_m v_m \\ &+2(\lambda_{m+1} - \lambda_m)^T(h_m + \frac{1}{2}\nabla h_m^T s_m) \\ &+\rho\ [\|h_m\|^2 - \|h_m + \nabla h_m^T s_m\|^2]. \end{aligned}$$

Using (5.4) and the fact that $2(Z_m^T\nabla f_m + Z_m^T B_m s_m^n)^T v_m + v_m^T Z_m^T B_m Z_m v_m \le 0$, we can write

$$\begin{aligned} r_m\|h_m\|\min[\frac{\tau\Delta_m}{b_0}, \|h_m\|] &\le 2\|\lambda_{m+1} - \lambda_m\|\|h_m + \frac{1}{2}\nabla h_m^T s_m\| \\ &\quad + \|Y_m^T B_m Z_m\|\|u_m\|\|v_m\| - 2\rho\, h_m^T \nabla h_m^T s_m. \end{aligned}$$

Using (4.1), (4.2), (4.3), (4.5) and the fact that $\|h_k + \frac{1}{2}\nabla h_k^T s_k\| \le \|h_k\|$, we can write

$$r_m\|h_m\|\min[\frac{\tau\Delta_m}{b_0}, \|h_m\|] \le (2b_1 + bb_0 + 2\rho b_3)\|h_m\|\|s_m\|.$$

If we use the above inequality together with the fact that r_k is updated by (3.6) only when $\|h_k\| > c_1\Delta_k$, we obtain

$$r_m \le \frac{\sqrt{2}b_0(2b_1 + bb_0 + 2\rho b_3)}{\min(\tau, c_1 b_0)}.$$

This result together with the fact that $\overline{\rho}_{m-1}$ does not satisfy (5.16) imply that $\overline{\rho}_m$ does not satisfy (5.16). This contradicts the supposition that m is the first index such that (5.16) is satisfied and means that there is no such m. Hence the sequence $\overline{\rho}_k$ is bounded. □

From the definition of $\{\overline{\rho}_k\}$ and the last lemma, it follows directly that the sequences $\{r_k\}$ and $\{\underline{\rho}_k\}$ are bounded.

LEMMA 5.9. *The sequence* $\{\underline{\rho}_k\}$ *is monotonically nondecreasing.*

Proof. From the way of updating the penalty parameter r_k we always have, for all k, $\underline{\rho}_{k-1} \le r_k$ and since $\underline{\rho}_k = \min\{r_k, r_{k-1}, ..., r_{k-N+1}\}$, then we must have $\underline{\rho}_{k-1} \le \underline{\rho}_k$ which means that the sequence $\{\underline{\rho}_k\}$ is monotonically non-decreasing. □

Now we argue that $\{\underline{\rho}_k\}$ will increase in a finite number of iterations until it reaches its upper bound. In other words, there exists an integer $\hat{k}$ such that $\underline{\rho}_k = \underline{\rho}_{\hat{k}}$ for all $k \ge \hat{k}$.

First of all, we study the possible increase in r_k over $\underline{\rho}_{k-1}$. In other words, if there is an increase in r_k over $\underline{\rho}_{k-1}$, how much is this increase?. If r_k is increased over $\underline{\rho}_{k-1}$, it will increase through one of the following three possibilities:

1) It will be increased by at least ρ if it is increased according to (3.6) regardless of the result in equation (3.5) of Scheme 3.4.
2) It will be increased by at least ρ if $\underline{\rho}_{k-1} + \rho \le \overline{\rho}_{k-1}$ regardless of the result in the "if" statement of Scheme 3.4.
3) It will be increased by at least $(\overline{\rho}_{k-1} - \underline{\rho}_{k-1})$ if $\underline{\rho}_{k-1} < \overline{\rho}_{k-1}$ but $\underline{\rho}_{k-1} + \rho > \overline{\rho}_{k-1}$.

Notice that the amount $(\overline{\rho}_{k-1} - \underline{\rho}_{k-1})$ can be very small so that, if at each iteration the penalty parameter increases by this amount, it seems that the algorithm may take infinitely many iterations without $\{\underline{\rho}_k\}$ reaching its upper bound. Later on we will show that this situation can not happen.

Also, we notice that, for $\underline{\rho}_{k-1} < \overline{\rho}_{k-1}$ we always have $\underline{\rho}_{k-1} < r_k$ which means a possible increase in $\underline{\rho}_{k-1}$ to $\underline{\rho}_k$.

Finally, we notice that, the only possibility that $r_k = \underline{\rho}_{k-1}$ is when $\underline{\rho}_{k-1} = \overline{\rho}_{k-1}$ and $\underline{\rho}_{k-1}$ satisfies (3.7). In this case $\underline{\rho}_{k-1} = r_k = \overline{\rho}_{k-1}$ which will imply $\underline{\rho}_k = r_k = \overline{\rho}_k$.

Define the following three sets of indices:

$I = \{\ k\ :\ \underline{\rho}_{k-1} + \rho \le \overline{\rho}_{k-1}\ \}$.

$J = \{ k \ : \ \underline{\rho}_{k-1} < \overline{\rho}_{k-1}$ but $\underline{\rho}_{k-1} + \rho > \overline{\rho}_{k-1} \}$.
$K = \{ k \ : \ \underline{\rho}_{k-1} = \overline{\rho}_{k-1} \}$.

The following propositions can be easily verified.

Proposition 1.
If $k+1 \in I$ then $\underline{\rho}_{k+N} \geq \underline{\rho}_k + \rho$.

Proposition 2.
If $k \in K$ then either $k+1 \in K$, or $k+1, ..., k+N-1 \in I$.

Proposition 3.
If $k \in J$ then either $k+1 \in J$, or $k+1 \in K$, or $k+1, ..., k+N-1 \in I$.

Proposition 4.
If $k, k+1, ..., k+N-1 \in J$, then $k+N \in K$, or $k+N, ..., k+2N-2 \in I$.

It is easy to see that (in the worst case) every $2N-1$ consecutive iterations at which the sequence $\{\underline{\rho}_k\}$ increases, it will increase by at least ρ. Thus, because $\{\underline{\rho}_k\}$ is bounded, the sequence $\{\underline{\rho}_k\}$ will take only a finite number of iterations to attain its upper bound.

LEMMA 5.10. *If the algorithm does not terminate, then there exists a positive integer k_2 and a constant $r_\star > 0$ such that, for all $k \geq k_2$, $r_k = r_\star$.*

Proof. We notice that, because of Lemma 5.8, after finite number of iterations k_1 inequality (3.7) will be satisfied for all $k \geq k_1$. This implies that there exists an integer $k_2 > k_1$ such that $\underline{\rho}_k = \overline{\rho}_k$ for all $k \geq k_2$. However, from the way of updating r_k, this will imply that $\underline{\rho}_k = r_k = \overline{\rho}_k$ for all $k \geq k_2$. This implies $r_k = r_\star$ for all $k \geq k_2$. □

5.3. The Main Global Results. We show that the algorithm always terminates. This is shown in two steps : First, it is shown that if the algorithm would not terminate, then $lim_{k\to\infty} \|h_k\| = 0$. Second, it is shown that if the algorithm would not terminate, then $\liminf_{k\to\infty} \|Z_k^T \nabla f_k\| = 0$. Thus for each $\epsilon > 0$ there exists k_0 such that $\|h_{k_0}\| + \|Z_{k_0}^T \nabla f_{k_0}\| < \epsilon$.

The following lemma is crucial in proving that the algorithm will converge to a feasible point. Intuitively speaking, it shows that the trust region will not collapse to a point as long as $\|h_k\|$ is bounded away from zero.

LEMMA 5.11. *Let the global assumptions hold. If the sequence of iterates generated by the algorithm is bounded away from the feasible region, i.e. $\|h_k\| > \varepsilon_0$, for some fixed positive constant ε_0 and and all k, then there exists a constant $c_3 > 0$, such that, for all k*

$$\Delta_k \geq c_3. \tag{5.17}$$

Proof. The proof is by contradiction. Suppose that $\{\Delta_k\}$ is not bounded away from zero, then there exists a sequence of indices $\{k_j\}$ such that

$$\Delta_k < \frac{a_1 b_0 \sigma_1}{\tau}(1-\eta_2), \tag{5.18}$$

for all $k \in \{k_j\}$, where $\sigma_1 = \min\{\varepsilon_0, \frac{\tau\Delta_1}{a_1 b_0 (1-\eta_2)}, \frac{\tau^2 \varepsilon_0}{2\sqrt{2} b_4 b_0^2}\}$.
Let m be the first integer such that (5.18) holds. It is clear from the definition of σ_1 that $\sigma_1 \leq \frac{\tau\Delta_1}{a_1 b_0 (1-\eta_2)}$ which implies that $m \geq 2$.
Using (5.18) and the way of updating Δ_k, we can write

$$\frac{\tau \|s_{m^j-1}\|}{\sqrt{2} b_0} \leq \frac{\tau \|s_{m^j-1}\|}{b_0} \leq \frac{\tau \Delta_m}{a_1 b_0} \leq \sigma_1 (1-\eta_2) \leq \sigma_1 \leq \varepsilon_0, \tag{5.19}$$

where s_{m^j-1} is the last rejected step, just before finding an acceptable one and moving to the point (x_m, λ_m). Here $s_{m^j-1} = s_{m-1}$ if there is no rejected ones between s_{m-1} and s_m. We obtain from (5.3), that

$$Pred_{m^j-1} \geq \frac{r_{m^j-1}\varepsilon_0}{2}\min\{\varepsilon_0, \frac{\tau\Delta_{m^j-1}}{b_0}\} \geq \frac{r_{m^j-1}\varepsilon_0\tau\|s_{m^j-1}\|}{2\sqrt{2}b_0}. \tag{5.20}$$

On the other hand, from (5.1),

$$|Ared_{m^j-1} - Pred_{m^j-1}| \leq r_{m^j-1}b_4\|s_{m^j-1}\|^2.$$

From (5.19), (5.20), and the above inequality, we have

$$\frac{|Ared_{m^j-1} - Pred_{m^j-1}|}{Pred_{m^j-1}} \leq \frac{2\sqrt{2}b_4b_0}{\tau\varepsilon_0}\|s_{m^j-1}\| \leq \frac{2\sqrt{2}b_4b_0^2\sigma_1(1-\eta_2)}{\tau^2\varepsilon_0} \leq (1-\eta_2).$$

The above inequality implies that the step s_{m^j-1} was an acceptable step. *i. e* $s_{m^j-1} = s_{m-1}$. It also implies that $\Delta_{m^j-1} \leq \Delta_m$ and means that Δ_{m^j-1} satisfies (5.18). This contradicts the supposition that m is the first integer such that (5.18) holds. Therefore, there is no integer k such that (5.18) holds. Hence the lemma is proved. □

The following theorem proves that under the global assumptions, either the algorithm satisfies its termination condition, or it converges to a feasible point.

THEOREM 5.12. *Let the global assumptions hold. If all members of the sequence of iterates generated by the algorithm fail to satisfy the termination condition, then*

$$\lim_{k\to\infty} \|h_k\| = 0.$$

Proof. We prove the theorem in two steps: First, we show that $\liminf_{k\to\infty}\|h_k\| = 0$, then we use this result to prove the theorem.

Assume there is an $\varepsilon_1 > 0$ such that $\|h_k\| \geq \varepsilon_1$, for all k. For any k, we have

$$\Phi_k - \Phi_{k+1} = Ared_k \geq \eta_1 Pred_k \geq \frac{\eta_1}{2}\|h_k\|\min[\frac{\tau\Delta_k}{b_0}, \|h_k\|]. \tag{5.21}$$

Since $\{\Phi_k\}$ is bounded below, $\Phi_{k+1} < \Phi_k$, for all $k \geq k_2$, where k_2 is as in Lemma 5.10 and $\|h_k\| \geq \varepsilon_1$ for all k, it follows that

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

On the other hand, because $\|h_k\| \geq \varepsilon_1$ for all k, Lemma 5.11 implies the existence of a constant $\bar{c}_3$, such that $\Delta_k > \bar{c}_3$ for all k which is a contradiction with the above limit.

Therefore, the assumption $\|h_k\| \geq \varepsilon_1$ for all k has led to a contradiction. Hence

$$\liminf_{k\to\infty} \|h_k\| = 0. \tag{5.22}$$

This result shows that at least one subsequence of $\{x_k\}$ will converge to a feasible point.

Now we will show that every subsequence will converge to a feasible point. Suppose that there exists a subsequence $\{k_j'\}$ of indices such that $\|h_{k_j'}\| > \varepsilon_1$. Because of this and (5.22) we may select two subsequences $\{k_j\}$ and $\{l_j\}$ as follows: Let $\{k_j\} \subset \{k'_j\}$ and for each j we select an l_j, such that

$$l_j = \max\{l \in [k_j, k_{j+1}) : \|h_i\| > \frac{\varepsilon_1}{2},\ k_j \leq i \leq l\}, \text{ and } \|h_{l_j+1}\| < \frac{\varepsilon_1}{2}.$$

From (5.21), for all iterates l such that $k_j \leq l \leq l_j, j = 1, 2, \ldots$, we have

$$\Phi_l - \Phi_{l+1} \geq \frac{\eta_1 \varepsilon_1}{4} \min[\frac{\tau \Delta_k}{b_0}, \frac{\varepsilon_1}{2}].$$

From the above inequality, it follows that

$$\Phi_{k_j} - \Phi_{l_j+1} = \sum_{l=k_j}^{l_j} (\Phi_l - \Phi_{l+1}) \geq \frac{\eta_1 \varepsilon_1}{4} \sum_{l=k_j}^{l_j} \min[\frac{\tau \Delta_l}{b_0}, \frac{\varepsilon_1}{4}].$$

This implies $\sum_{l=k_j}^{l_j} \Delta_l \to 0$. But

$$\sum_{l=k_j}^{l_j} \Delta_l \geq \sum_{l=k_j}^{l_j} \frac{\|s_l\|}{\sqrt{2}} \geq \frac{1}{2} \|x_{k_j} - x_{l_j+1}\|.$$

So, as $j \to \infty$, $\|x_{k_j} - x_{l_j+1}\| \to 0$. This implies that there exists an integer k_3 sufficiently large such that $\|x_{k_j} - x_{l_j+1}\| \leq \frac{\varepsilon_1}{2\gamma}$, where $\gamma = \max(b_2, 1)$. Now, using (4.4), we have

$$\|h_{k_j}\| \leq \|h_{k_j} - h_{l_j+1}\| + \|h_{l_j+1}\| \leq \frac{b_2 \varepsilon_1}{2\gamma} + \frac{\varepsilon_1}{2} \leq \varepsilon_1$$

for all k_j sufficiently large which is a contradiction.
So the supposition that $\|h_{k'_j}\| > \varepsilon_1$ has led to a contradiction. Hence, the supposition is wrong and the theorem is proved. □

The following lemma is needed in the proof of Theorem 5.14. It proves that under the assumption that the algorithm does not terminate, if $\{\|Z_k^T \nabla f_k\|\}$ is bounded away from zero, then the trust-region radius will be bounded away from zero.

LEMMA 5.13. *Let the global assumptions hold. If all members of the sequence of iterates generated by the algorithm fail to satisfy the termination condition and satisfy* $\|Z_k^T \nabla f_k\| > \varepsilon_2$, *for some fixed constant* $\varepsilon_2 > 0$, *then*

$$\Delta_k \geq c_4 \tag{5.23}$$

where c_4 *is a positive constant independent on* k.

Proof. Since the algorithm does not terminate, then from Theorem 5.12, $\|h_k\| \to 0$. Hence there exists k_4 sufficiently large, such that, for all $k \geq k_4$, we have

$$\|h_k\| \leq \min\{\frac{\varepsilon_2}{2bb_0}, \frac{\varepsilon_2}{16\sqrt{2}b_5} \min[1, \frac{\varepsilon_2}{4b\Delta_*}]\}. \tag{5.24}$$

Now, using (5.24), we can write

$$\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \geq \|Z_k^T \nabla f_k\| - bb_o \|h_k\| \geq \frac{\varepsilon_2}{2}.$$

From Lemma 5.3, Lemma 5.4, and the above inequality, we can write

$$Pred_k \geq \frac{1}{2} Tpred_k + \{\frac{\varepsilon_2}{16}\min[1, \frac{\varepsilon_2}{4b\Delta_*}] - \sqrt{2}b_5\|h_k\|\}\Delta_k.$$

Again, by using (5.24), we obtain

$$Pred_k \geq \frac{1}{2} Tpred_k \geq \frac{1}{8}\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|\min[\Delta_k, \frac{\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|}{2b}].$$

Hence, for all $k \geq k_4$, we have

$$Pred_k \geq \frac{\varepsilon_2}{16}\min[\Delta_k, \frac{\varepsilon_2}{4b}]. \tag{5.25}$$

The rest of the proof is by contradicting (5.23). Suppose that $\{\Delta_k\}$ is not bounded away from zero. Then there exists a sequence of indices $\{k_j\}$ such that

$$\Delta_k < a_1\sigma_2(1-\eta_2), \tag{5.26}$$

for all $k \in \{k_j\}$, where

$$\sigma_2 = \min\{\frac{\varepsilon_2}{4b}, \frac{\varepsilon_2}{16\sqrt{2}r_* b_4}, \frac{\Delta_{k_4}}{a_1(1-\eta_2)}\}.$$

Let m be the first integer such that (5.26) holds. It is clear that $m \geq k_4 + 1$. Using (5.26), then from the way of updating Δ_k, we can write

$$\frac{\|s_{m^j-1}\|}{\sqrt{2}} \leq \|s_{m^j-1}\| \leq \frac{\Delta_m}{a_1} < \sigma_2(1-\eta_2) \leq \sigma_2 \leq \frac{\varepsilon_2}{4b}, \tag{5.27}$$

where $s_m^j - 1$ is the last rejected step, just before finding an acceptable one and moving to the point (x_m, λ_m). Observe that $s_m^j - 1 = s_{m-1}$ if there is no rejected ones between s_{m-1} and s_m. We obtain from (5.25) and (5.27), that

$$Pred_{m^j-1} \geq \frac{\varepsilon_2}{16\sqrt{2}}\|s_{m^j-1}\| \tag{5.28}$$

From (5.1), we have

$$|Ared_{m^j-1} - Pred_{m^j-1}| \leq r_* b_4 \|s_{m^j-1}\|^2$$

By using the above inequality and (5.28), we obtain

$$\frac{|Ared_{m^j-1} - Pred_{m^j-1}|}{Pred_{m^j-1}} \leq \frac{16\sqrt{2}r_* b_4}{\varepsilon_2}\|s_{m^j-1}\| \leq \frac{16\sqrt{2}r_* b_4\sigma_2(1-\eta_2)}{\varepsilon_2} \leq (1-\eta_2).$$

The above inequality implies that the step s_{m^j-1} was an acceptable one. *i. e* $s_{m^j-1} = s_{m-1}$. It also implies that $\Delta_{m^j-1} \leq \Delta_m$ and means that $m-1$ satisfies (5.26). This contradicts the supposition that m is the first integer such that (5.26) holds. Therefore, there is no integer k such that (5.26) holds. Hence the lemma is proved. □

The following theorem proves that under the global assumptions, if each member of the sequence of iterates generated by the algorithm does not satisfy the termination condition of the algorithm, then there exists a subsequence $\{x_{k_j}\}$ of these iterates for which $\{||Z_{k_j}^T \nabla f_{k_j}||\}$ converges to zero.

THEOREM 5.14. *Let the global assumptions hold. If all members of the sequence of iterates generated by the algorithm fail to satisfy the termination condition , then*

$$\liminf_{k\to\infty} ||Z_k^T \nabla f_k|| = 0.$$

Proof. The proof is by contradiction. Suppose that there exists an $\varepsilon_3 > 0$ such that $||Z_k^T \nabla f_k|| \geq \varepsilon_3$ for all k. As in Lemma 5.13, there exists k_4 sufficiently large such that for all $k \geq k_4$, we have

$$Pred_k \geq \frac{\varepsilon_3}{16}\min[\Delta_k, \frac{\varepsilon_3}{4b}].$$

On the other hand, for all $k \geq k_2$, $r_k = r_\star$. Hence, for $k \geq \max\{k_4, k_2\}$, we have

$$\Phi_k - \Phi_{k+1} = Ared_k \geq \eta_1 Pred_k \geq \frac{\eta_1 \varepsilon_3}{16}\min[\Delta_k, \frac{\varepsilon_3}{4b}]. \tag{5.29}$$

Since Φ_k is bounded below and $\Phi_{k+1} < \Phi_k$, for all $k \geq \max\{k_4, k_2\}$, we have

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

On the other hand, because of the assumption that the algorithm does not terminate and that $||Z_k^T \nabla f_k|| \geq \varepsilon_3$, for all k, Lemma 5.13 implies the existence of a constant $\bar{c}_4$, such that $\Delta_k > \bar{c}_4$ for all k. This contradicts the above limit. Therefore, the supposition $||Z_k^T \nabla f_k|| \geq \varepsilon_3$, for all k has led to a contradiction. Hence the supposition is wrong and the lemma is proved. □

The above two theorems imply that under the global assumptions and the assumption that the algorithm does not terminate, the algorithm produces an infinite sequence of iterates $\{x_k\}$ that satisfies

$$\liminf_{k\to\infty} [||h_k|| + \ ||Z_k^T \nabla f_k||] = 0. \tag{5.30}$$

This result contradicts the assumption that the algorithm does not terminate and means that the termination condition of the algorithm will be met after finitely many iterations.

Satisfying the termination condition by itself means that the point at which the algorithm terminates lies in a ball of radius ε and center at a stationary point $(x_\star, \lambda_\star)$.

In practice there is no difference between $\liminf_{k\to\infty} [\ ||h_k|| + \ ||Z_k^T \nabla f_k||\ ] = 0$ and $\lim_{k\to\infty} [\ ||h_k|| + \ ||Z_k^T \nabla f_k||\ ] = 0$. Both mean that the algorithm will terminate after finitely many iterations.

If the point $(x_\star, \lambda_\star)$ is not an isolated local minimizer that satisfies the second order sufficiency condition, then our analysis is stopped here. On the other hand, if the algorithm avoids the neighborhoods of stationary points that do not satisfy the second order sufficiency condition, then we remove the termination condition from the algorithm and proceed, in the following section, with the local analysis.

6. The Local Analysis. In this section, in addition to the global assumptions, we add the following assumption:

Local Assumption A:

We assume that the problem has a finite number of isolated local minimizers and each one satisfies the second order sufficiency condition.

We remove the termination condition from the algorithm and proceed with the analysis. Because there is no termination condition, Lemma 5.10 and theorems 5.12 and 5.14 are no longer valid. However, the global analysis still imply that given any $\varepsilon > 0$ there exists a ball $\mathcal{B}_\varepsilon(\bar{x}, \bar{\lambda})$ of radius ε and center $(\bar{x}, \bar{\lambda})$, where $(\bar{x}, \bar{\lambda})$ is a stationary point of the problem, such that the sequence of iterates generated by the algorithm is not bounded away from this ball. *i. e.* for some k sufficiently large, we have $(x_k, \lambda_k) \in \mathcal{B}_\varepsilon(\bar{x}, \bar{\lambda})$.

The local analysis of our algorithm is presented in three sections. In Section 6.1 we study the behavior of the penalty parameter after removing the termination condition from the algorithm. In Section 6.2, we prove that the sequence of iterates $\{(x_k, \lambda_k)\}$ converges to a local minimizer $(x_\star, \lambda_\star)$. Section 6.3 is devoted to studying the local rate of convergence of our algorithm. We show that our globalization strategy will not disrupt the fast local rate of convergence.

If the point $(x_\star, \lambda_\star)$ satisfies the second order sufficiency condition (see Section 1), then by the continuity assumption, there exists a neighborhood $\mathcal{N}(x_\star, \lambda_\star)$ of $(x_\star, \lambda_\star)$ such that $Z(x)^T \nabla_x^2 l(x, \lambda) Z(x) > 0$, for all $(x, \lambda) \in \mathcal{N}(x_\star, \lambda_\star)$.

6.1. The Local Behavior of The Penalty Parameter. In this section, we prove technical lemmas needed to study the local behavior of the penalty parameter. At the end of this section we prove that, under the Global Assumptions and Assumption A, the penalty parameter is bounded.

The point $(x_\star, \lambda_\star)$ is used in this section to mean a stationary point of the problem that satisfies the second order sufficiency condition and $\mathcal{N}(x_\star, \lambda_\star)$ is used to mean a neighborhood of $(x_\star, \lambda_\star)$ such that $Z(x)^T \nabla_x^2 l(x, \lambda) Z(x) > 0$, for all $x \in \mathcal{N}(x_\star, \lambda_\star)$.

LEMMA 6.1. *If $(x_k, \lambda_k) \in \mathcal{N}(x_\star, \lambda_\star)$, there exists a positive constant e_1, such that*

$$\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \geq e_1 \|v_k\|.$$

Proof. Since $(x_k, \lambda_k) \in \mathcal{N}(x_\star, \lambda_\star)$ then $Z_k^T B_k Z_k$ is positive definite. Hence, there exists a positive constant e_1 such that, for all k sufficiently large $e_1 \|v_k\|^2 \leq v_k^T Z_k^T B_k Z_k v_k$. Now, since

$$v_k^T Z_k^T B_k Z_k v_k \leq -(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k,$$

we can write

$$e_1 \|v_k\| \leq \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|. \tag{6.1}$$

This completes the proof. □

LEMMA 6.2. *If $(x_k, \lambda_k) \in \mathcal{N}(x_\star, \lambda_\star)$ is such that $\|h_k\| \leq e_2 \|s_k\|$ where $0 < e_2 \leq \frac{1}{2b_0}$ and b_0 is as in (4.2), then*

$$\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \geq \frac{e_1}{2} \|s_k\|.$$

Proof. Since $\|u_k\| + \|v_k\| \geq \|s_k\|$, then by using (4.2) and (6.1), we obtain

$$e_1 b_0 \|h_k\| + \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \geq e_1 \|s_k\|.$$

When $\|h_k\|_2 \leq e_2 \|s_k\|_2$, we have

$$\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|_2 \geq e_1 (1 - e_2 b_0) \|s_k\|.$$

Using $e_2 \leq \frac{1}{2b_0}$, we obtain the desired result. □

LEMMA 6.3. *If $(x_k, \lambda_k) \in \mathcal{N}(x_\star, \lambda_\star)$ is such that $\|h_k\| \leq e_3\|s_k\|$ where e_3 is a positive constant that satisfies:*

$$e_3 \leq \min[e_2, \frac{e_1 \min(4b, \sqrt{2}e_1)}{64\sqrt{2}bb_5}], \tag{6.2}$$

where b is as in (4.1), b_5 is as in Lemma 5.4, e_1 is as in Lemma 6.1, and e_2 is as in Lemma 6.2, then

$$Pred_k \geq \frac{1}{2}Tpred_k + \frac{r_k}{2}Npred_k. \tag{6.3}$$

Proof. From Lemma 5.3 and Lemma 5.4, we have

$$\begin{aligned} Pred_k &\geq \frac{1}{2}Tpred_k + \frac{1}{8}\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|\min[\Delta_k, \frac{\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|}{2b}] \\ &\quad -b_5\|s_k\|\|h_k\| + \frac{r_k}{2}Npred_k. \end{aligned} \tag{6.4}$$

Now, since $\|h_k\| \leq e_3\|s_k\|$ and $e_3 \leq e_2$ then by using Lemma 6.2 we have $\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\| \geq \frac{e_1}{2}\|s_k\|$, and using (6.2), we obtain

$$\begin{aligned} \frac{1}{8}\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\| \quad \min \quad &[\Delta_k, \frac{\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|}{2b}] - b_5\|s_k\|\|h_k\| \\ \geq \quad &\{\frac{e_1}{16}\min[\frac{1}{\sqrt{2}}, \frac{e_1}{4b}] - b_5 e_3\}\|s_k\|^2 \geq 0. \end{aligned}$$

The remainder of the proof follows immediately. □

From the proof of the above lemma we see that, if $\|h_k\| \leq e_3\|s_k\|$, then the second term in (6.4) will cancel the third term and the fourth term need never enter into the calculation. This implies that if we set $r_k = \rho_{k-1}$, (see Scheme 3.4), inequality (6.3) remains valid. In this case the algorithm will not update r_k using (3.6) because inequality (3.7) will be satisfied.

LEMMA 6.4. *If for all k, $(x_k, \lambda_k) \in \mathcal{N}(x_\star, \lambda_\star)$, then $r_k \leq r^\star$, where $r^\star$ is a positive constant that does not depend on k.*

Proof. First we follow a proof similar to the proof of Lemma 5.8. We demonstrate the boundedness of the sequence $\{\overline{\rho}_k\}$. The rest of the proof follows because, for all k, $r_k \leq \overline{\rho}_k$. □

LEMMA 6.5. *Under the global and the local assumptions, the sequence r_k is bounded.*

Proof. Because we have a finite number, p say, of local minimizers that satisfies the second order sufficiency condition (see Assumption A), we can find a radius $\bar{\varepsilon}$, such that $\mathcal{B}^i_{\bar{\varepsilon}}(x^i_\star, \lambda^i_\star) \subset \mathcal{N}^i(x^i_\star, \lambda^i_\star)$, for $i = 1, 2, ..., p$.

Now consider the set $\mathcal{B}_{\bar{\varepsilon}} = \bigcup_{i=1}^p \mathcal{B}^i_{\bar{\varepsilon}}(x^i_\star, \lambda^i_\star)$. If any iterate k is such that $(x_k, \lambda_k) \notin \mathcal{B}_{\bar{\varepsilon}}$ then from the global analysis, there exists a constant $\bar{r}_\star$ such that, $r_k \leq \bar{r}_\star$. Observe that $\bar{r}_\star$ depends on $\bar{\varepsilon}$. Here $\bar{\varepsilon}$ is fixed. On the other hand, if $(x_k, \lambda_k) \in \mathcal{B}_{\bar{\varepsilon}}$ then from Lemma 6.4, there exists a constant $\bar{r}^\star$, such that $r_k \leq \bar{r}^\star$. Now take $\bar{r} = \max(\bar{r}_\star, \bar{r}^\star)$, we can see that the sequence $\{r_k\}$ is bounded by $\bar{r}$. □

Now we follow the argument that comes immediately after the proof of Lemma 5.9, and then follow the proof of Lemma 5.10, we conclude that there exists an integer $\overline{k}$ such that for all $k \geq \overline{k}$ the sequence of penalty parameters reaches its upper bound.

In the following section we study the sequence of points $\{(x_k,\lambda_k)\}$ generated by the algorithm after the penalty parameter reaches its upper bound.

Without loss of generality we may assume that the sequence of penalty parameters is independent of k.

6.2. First Order Convergence. From the global analysis, there exists a subsequence of points $\{(x_{k_j},\lambda_{k_j})\}$ generated by the algorithm, such that $(x_k,\lambda_k)\in\mathcal{N}(x_*,\lambda_*)$, for all $k\in\{k_j\}$.

Consider the level sets $\mathcal{L}_k\equiv\{(x,\lambda):\Phi(x,\lambda,r)\le\Phi(x_k,\lambda_k,r)\}$. There exists an integer $\hat{k}$ sufficiently large, such that $\mathcal{L}_{\hat{k}}\subset\mathcal{N}(x_*,\lambda_*)$.

The following lemma proves that there exists an index $\bar{k}$ such that all the subsequent iterates generated by the algorithm will never leave the level set $\mathcal{L}_{\bar{k}}$.

LEMMA 6.6. *Under the global and the local assumptions, there exists an index $\bar{k}$ sufficiently large, such that $(x_k,\lambda_k)\in\mathcal{L}_{\bar{k}}$, for all $k>\bar{k}$.*

Proof. From the global analysis and the local assumption A, there exists an index $\bar{k}$ such that $\mathcal{L}_{\bar{k}}\subset\mathcal{B}_{\bar{\epsilon}}$.

The proof now is by contradiction. Suppose that some iterates leave the set $\mathcal{L}_{\bar{k}}$. Let $m+1$ be the first iterate that leaves the set. Therefore, $(x_m,\lambda_m)\in\mathcal{L}_{\bar{k}}$ and $(x_{m+1},\lambda_{m+1})\notin\mathcal{L}_{\bar{k}}$. Since s_m is an acceptable step, then we have

$$\Phi_m-\Phi_{m+1}=Ared_m\ge\eta_1 Pred_m\ge 0.$$

Then $\Phi_m\ge\Phi_{m+1}$. This implies that $(x_{m+1},\lambda_{m+1})\in\mathcal{L}_{\bar{k}}$. This gives a contradiction. Hence the lemma is proved. □

THEOREM 6.7. *Under the global and the local assumptions, the algorithm will generate points that satisfy*

$$\lim_{k\to\infty}\|h_k\|=0.$$

Proof. The proof is similar to the proof of Theorem 5.12. □

THEOREM 6.8. *Under the global and the local assumptions, we have*

$$\lim_{k\to\infty}\|Z_k^T\nabla f_k\|=0.$$

Proof. First we follow a proof similar to the proof of theorem 5.14. We demonstrate

$$\liminf_{k\to\infty}\|Z_k^T\nabla f_k\|=0. \tag{6.5}$$

The rest of the proof will follow by contradiction. Suppose there exists a subsequence of indices $\{k_j\}$ such that $k_j\ge\bar{k}$, where $\bar{k}$ is as in Lemma 6.6, and $\|Z_k^T\nabla f_k\|>\sigma_1$ for all $k\in\{k_j\}$, where $\sigma_1>0$.

Take an iterate $k'\in\{k_j\}$ sufficiently large such that for all $k\ge k'$, we have

$$\|h_k\|\le\min\{\frac{\sigma_1}{2bb_0},\frac{\sigma_1}{16\sqrt{2}b_5}\min[1,\frac{\sigma_1}{4b\Delta_*}]\}. \tag{6.6}$$

For some $\beta>o$ and any $x\in\Omega$, we have

$$\begin{aligned}\|Z(x)^T\nabla f(x)\| &\ge \|Z_{k'}^T\nabla f_{k'}\|-\|Z(x)^T\nabla f(x)-Z_{k'}^T\nabla f_{k'}\|\\ &\ge \|Z_{k'}^T\nabla f_{k'}\|-\beta\|x-x_{k'}\|.\end{aligned}$$

This implies that $\|Z(x)^T\nabla f(x)\| \geq \frac{1}{2}\|Z_{k'}^T\nabla f_{k'}\| > \frac{\sigma_1}{2}$ holds for every $x \in \Omega$ that satisfies

$$\|x - x_{k'}\| \leq \frac{\|Z_{k'}^T\nabla f_{k'}\|}{2\beta}.$$

Therefore, take $\sigma_2 = \frac{\|Z_{k'}^T\nabla f_{k'}\|}{2\beta}$ and consider the ball $U_{\sigma_2} = \{x : \|x - x_{k'}\| \leq \sigma_2\}$. For all $k \geq k'$ such that $x_k \in U_{\sigma_2}$, we have $\|Z_k^T\nabla f_k\| > \frac{\sigma_1}{2}$. As in Lemma 5.13 (because of (6.6)), we have, for all $k \geq k'$

$$\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\| > \frac{\sigma_1}{4}$$

and

$$Pred_k \geq \frac{1}{8}\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\| \min\{\Delta_k, \frac{\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|}{2b}\}.$$

This implies that for any iterate $k \geq k'$ that lies inside the ball, we have

$$Pred_k > \frac{\sigma_1}{32}\min[\Delta_k, \frac{\sigma_1}{8b}].$$

Because of (6.5), the iterates, for all $k \geq k'$, can not stay in this ball. Let $l+1$ be the first integer greater than k' such that the point x_{l+1} does not lie inside the ball $\mathcal{U}_{\sigma_2}$. Hence,

$$\begin{aligned} \Phi_{k'} - \Phi_{l+1} &= \sum_{k=k'}^{l}(\Phi_k - \Phi_{k+1}) \geq \sum_{k=k'}^{l}\eta_1 Pred_k \\ &\geq \sum_{k=k'}^{l}\frac{\eta_1\sigma_1}{32}\min[\Delta_k, \frac{\sigma_1}{8b}]. \end{aligned}$$

Therefore,

$$\Phi_{k'} - \Phi_{l+1} \geq \frac{\eta_1\sigma_1}{32}\min[\frac{\sigma_2}{\sqrt{2}}, \frac{\sigma_1}{8b}]. \tag{6.7}$$

Since Φ_k is bounded below and is a decreasing sequence, $\{\Phi_k\}$ converges to some limit Φ_*. Taking the limit as l goes to infinity in inequality (6.7), we obtain

$$\Phi_{k'} - \Phi_* \geq \frac{\eta_1\sigma_1}{32}\min[\frac{\sigma_2}{\sqrt{2}}, \frac{\sigma_1}{8b}].$$

If we now take the limit as k' goes to infinity, we obtain

$$0 \geq \frac{\eta_1\sigma_1}{32}\min[\frac{\sigma_2}{\sqrt{2}}, \frac{\sigma_1}{8b}],$$

which contradicts the fact that $\sigma_1 > 0$ and $\sigma_2 > 0$. Hence there is no such sequence and the lemma is proved. □

6.3. The Local Rate of Convergence. In this section we prove Lemma 6.9 which is needed in our analysis. Then we prove Lemma 6.10 which proves that under the global and the local assumptions, for k sufficiently large, all the trial steps will be accepted and the trust region will not be decreased. In Theorems 6.11 and 6.12, we study the local rate of convergence of our algorithm. We show that asymptotically the trust region will be inactive and hence the fast local rate of convergence will be maintained.

LEMMA 6.9. *Under the global and the local assumptions, there exists a positive constant e_4 independent of k such that*

$$Pred_k \geq e_4 \|s_k\|^2.$$

Proof. If $\|h_k\| \leq e_3\|s_k\|$, where e_3 is as in (6.2), then using Lemmas 6.3 and 5.3

$$Pred_k \geq \frac{1}{2} Tpred_k \geq \frac{1}{8}\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \min[\frac{\|s_k\|}{\sqrt{2}}, \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{2b}].$$

But, since $\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \geq \frac{e_1}{2}\|s_k\|$, then

$$Pred_k \geq \frac{e_1}{16}\min[\frac{1}{\sqrt{2}}, \frac{e_1}{4b}]\|s_k\|^2.$$

On the other hand, when $\|h_k\| > e_3\|s_k\|$, from (5.3) and the fact that $r \geq \underline{\rho}_0 = 1$, we have

$$Pred_k \geq \frac{r}{2}\|h_k\|\min[\frac{\tau\Delta_k}{b_0}, \|h_k\|] \geq \frac{e_3}{2}\min[\frac{\tau}{\sqrt{2}b_0}, e_3]\|s_k\|^2.$$

If we take $e_4 = \min\{\frac{e_1}{64\sqrt{2}b}\min[4b, \sqrt{2}e_1], \frac{e_3}{2\sqrt{2}b_0}\min[\tau, \sqrt{2}b_0 e_3]\}$, we obtain the desired result. □

We add to our local assumptions the following set of assumptions:

Local Assumption B:

$\nabla_x^2 l$ is Lipschitz continuous in a neighborhood of the solution $x_\star$.

Local Assumption C:

If an approximation to the exact Hessian is used, then for all k, B_k satisfies:

$$\lim_{k\to\infty} \frac{\|Z_k(B_k - \nabla_x^2 l_\star)s_k\|}{\|s_k\|} = 0. \tag{6.8}$$

The above assumption is the Boggs-Tolle-Wang characterization of q-superlinear convergence of $\{x_k\}$ to $x_\star$. It is proved by Boggs, Tolle, and Wang (1982)[1] and Powell (1983)[17] that under the local assumptions, Algorithm 1.1 converges to $x_\star$ q-superlinearly. On the other hand, if the exact Hessian is used, the local convergence rate is q-quadratic. (See Goodman (1985)[10]).

The following Lemma shows that, for all k large enough, the trust-region radius Δ_k will not be decreased. *i.e.* the sequence $\{\Delta_k\}$, for k large enough, will form a non-decreasing sequence.

LEMMA 6.10. *Under the global and the local assumptions, there exists an integer k_5 sufficiently large, such that for all $k \geq k_5$, we have*

$$\frac{Ared_k}{Pred_k} \geq \eta_2.$$

Proof. We have, using (2.1),

$$\begin{aligned}\Phi(x_k+s_k,\lambda_{k+1},r) &= \Phi(x_k,\lambda_{k+1},r)+\nabla_x\Phi(x_k,\lambda_{k+1},r)^T s_k\\ &\quad +\frac{1}{2}s_k^T\nabla_x^2\Phi(x_k,\lambda_{k+1},r)^T s_k+o(\|s_k\|^2)\\ &= \Phi(x_k,\lambda_k,r)+\nabla_x\Phi(x_k,\lambda_k,r)^T s_k+\frac{1}{2}s_k^T\nabla_x^2\Phi(x_k,\lambda_k,r)^T s_k\\ &\quad +(\Delta\lambda_k)^T h_k+(\Delta\lambda_k)^T\nabla h_k^T s_k+\frac{1}{2}s_k^T(\Delta\lambda_k)^T\nabla^2 h_k s_k+o(\|s_k\|^2).\end{aligned}$$

From the above equation, using the definition of $Ared_k$, we obtain

$$\begin{aligned}Ared_k &= -\nabla_x l(x_k,\lambda_k)^T s_k-\frac{1}{2}s_k^T\nabla_x^2 l(x_k,\lambda_k)s_k\\ &\quad -(\Delta\lambda_k)^T(h_k+\nabla h_k^T s_k)-r[\|h_k+\nabla h_k^T s\|^2-\|h_k\|^2]\\ &\quad -\frac{1}{2}s_k^T\nabla^2 h_k\Delta\lambda_k s_k-rs_k^T\nabla^2 h_k h_k s_k-o(\|s_k\|^2).\end{aligned}$$

If we use the definition of $Pred_k$ and the above inequality, we obtain

$$\begin{aligned}Ared_k &\geq Pred_k-o(\|s_k\|^2)-r|s_k^T\nabla^2 h_k h_k s_k|-\frac{1}{2}s_k^T[\nabla_x^2 l_*-B_k]Z_k v_k\\ &\quad -\frac{1}{2}s_k^T[\nabla_x^2 l_k-\nabla_x^2 l_*]s_k-\frac{\alpha_k}{2}s_k^T\nabla_x^2 l_* Y_k u_k-\frac{1}{2}(\Delta\lambda_k)^T\nabla h_k^T s_k.\end{aligned}$$

We show first that the last two terms are $o(\|s_k\|^2)+o(\|s_k\|\|h_k\|)$. By differentiating the normal equation $Y(x)^T[\nabla_x l(x,\lambda(x))]=0$ at $x=x_*$, we obtain $Y_*^T[\nabla_x^2 l_*+\nabla h_*\nabla\lambda_*^T]=0$, or equivalently $R_*\nabla\lambda_*^T=-Y_*^T\nabla_x^2 l_*$.Therefore,

$$\begin{aligned}-\frac{1}{2}(\Delta\lambda_k)^T\nabla h_k^T s_k &= -\frac{\alpha_k}{2}s_k^T\nabla\lambda_k R_k^T u_k+o(\|s_k\|^2)\\ &= -\frac{\alpha_k}{2}s_k^T[\nabla\lambda_k R_k^T-\nabla\lambda_* R_*^T]u_k-\frac{\alpha_k}{2}s_k^T\nabla\lambda_* R_*^T u_k+o(\|s_k\|^2)\\ &= -\frac{\alpha_k}{2}s_k^T\nabla\lambda_* R_*^T u_k+o(\|s_k\|^2)+o(\|s_k\|\|h_k\|).\end{aligned}$$

Hence,

$$\begin{aligned}-\frac{\alpha_k}{2}s_k^T\nabla_x^2 l_* Y_k u_k-\frac{1}{2}(\Delta\lambda_k)^T\nabla h_k^T s_k &= -\frac{\alpha_k}{2}s_k^T\nabla_x^2 l_*[Y_*-Y_k]u_k+o(\|s_k\|^2)\\ &\quad +o(\|s_k\|\|h_k\|),\\ &= o(\|s_k\|^2)+o(\|s_k\|\|h_k\|).\end{aligned}$$

Using Lemma 6.4, for k large enough, we have

$$\begin{aligned}\frac{Ared_k}{Pred_k} &\geq 1-\frac{1}{e_4}[\frac{o(\|s_k\|^2)}{\|s_k\|^2}+\frac{o(\|s_k\|\|h_k\|)}{\|s_k\|^2}+\frac{r|s_k^T\nabla^2 h_k h_k s_k|}{\|s_k\|^2}\\ &\quad +\frac{|s_k^T[\nabla_x^2 l_*-B_k]Z_k v_k|}{2\|s_k\|^2}+\frac{|s_k^T[\nabla_x^2 l_k-\nabla_x^2 l_*]s_k|}{2\|s_k\|^2}].\end{aligned}$$

Using the local assumptions, Theorem 6.7, Theorem 6.8, Lemma 6.1 and inequality (4.2), we conclude that there exists an integer k_5 sufficiently large such that, for all $k\geq k_5$, we have

$$\frac{Ared_k}{Pred_k}\geq\eta_2.$$

Hence, the theorem is proved. □

In our definition of $Pred_k$ we used $\frac{1}{2}s_k^T B_k Z_k v_k$ instead of $\frac{1}{2}s_k^T B_k s_k$ and used $h_k + \frac{1}{2}\nabla h_k^T s_k$ instead of $h_k + \nabla h_k^T s_k$. This way of defining $Pred_k$ allows us, when comparing with the second order approximation of the terms of $Ared_k$, to have two extra terms, namely, $\frac{1}{2}s_k^T B_k Y_k u_k$ and $\frac{1}{2}\nabla h_k^T s_k$. These two terms are very important in our local analysis because they allow us, using the Local Assumption C, to prove that $Pred_k$ approximates $Ared_k$ to within terms that are of order $o(\|s_k\|^2)$ or $o(\|s_k\|\|h_k\|)$.

Now as $k \to 0$, $\|h_k\| \to 0$ and $\|Z_k^T \nabla f_k\| \to 0$ and hence $\|s_k\| \to 0$. This implies that $\frac{Ared_k}{Pred_k} \to 1$, which means that for k sufficiently large all the steps produced by our algorithm are acceptable. This also means that for k sufficiently large the sequence of trust region radii $\{\Delta_k\}$ is a non-decreasing sequence.

The following two theorems show that the fast local rate of convergence will be maintained.

THEOREM 6.11. *Under the global and the local assumptions, if the exact Hessian is used, then for k sufficiently large, $x_k \to x_\star$ q-quadraticly.*

Proof. From Lemma 6.10, the trust region radius Δ_k for $k \geq k_5$ is updated according to the rule $\Delta_{k+1} = \min\{\Delta_\star, \max[\Delta_k, a_3\|s_k\|]\}$. Hence, $\Delta_k \geq \Delta_{k_5}$ for all $k \geq k_5$. However, for all k, $\Delta_k \leq \Delta_\star$.

First, we show that the trust region will be inactive, for sufficiently large k. Suppose there exists an integer $k_6 \geq k_5$ such that the full normal and tangential components of the step are not taken for all $k \geq k_6$. This implies that, for all $k \geq k_6$, $\|R_k^{-T} h_k\| = \|u_k\| > \Delta_k \geq \Delta_{k_6}$ and $\|(Z_k^T B_k Z_k)^{-1}(Z_k^T \nabla l_k + Z_k^T B_k s_k^n)\| > \|v_k\| = \Delta_k \geq \Delta_{k_6}$. But, using (4.2) and Lemma 6.1, this will contradict the fact that $\|h_k\| \to 0$ and $\|Z_k^T \nabla f_k\| \to 0$. Therefore, there exists a subsequence of indices $\{k_j\}$ such that $\|s_{k_j}^t\| \leq \Delta_{k_6}$ and $\|s_{k_j}^n\| \leq \Delta_{k_6}$ where all of $k_j \geq k_6$.

Let $m \in \{k_j\}$ be the smallest integer greater than k_6 such that $\|s_m^t\| \leq \Delta_{k_6}$, $\|s_m^n\| \leq \Delta_{k_6}$, and such that the local method, *i.e.* Algorithm 1.1, generates steps that are q-quadratic, *i.e.* satisfies

$$\|s_{k+1}\| \leq \beta_1 \|s_k\|^2,$$

where β_1 is a constant. But since the local method converges r-quadratic in the components s_k^t and s_k^n. This implies the existence of an integer $k_7 \geq m$, such that for all $k \geq k_7$, we have

$$\|R_k^{-T} h_k\| \leq \beta_2 (\gamma_1^2)^{k_7}$$

and

$$\|(Z_k^T B_k Z_k)^{-1}(Z_k^T \nabla l_k + Z_k^T B_k s_k^n)\| \leq \beta_3 (\gamma_2^2)^{k_7},$$

where $\beta_2, \beta_3, \gamma_1$, and γ_2 are constants and $\gamma_1, \gamma_2 \in (0,1)$.

This means that if we choose k_7 sufficiently large such that $\max\{\beta_2(\gamma_1^2)^{k_7}, \beta_3(\gamma_2^2)^{k_7}\} \leq \Delta_{k_7}$ then we have, $\|R_{k_7}^{-T} h_{k_7}\| \leq \Delta_{k_7}$, $\|(Z_{k_7}^T B_{k_7} Z_{k_7})^{-1}(Z_{k_7}^T \nabla l_{k_7} + Z_{k_7}^T B_{k_7} s_{k_7}^n)\| \leq \Delta_{k_7}$, and for all $k \geq k_7$, we have

$$\|R_k^{-T} h_k\| \leq \Delta_{k_7},$$

and

$$\|(Z_k^T B_k Z_k)^{-1}(Z_k^T \nabla l_k + Z_k^T B_k s_k^n)\| \leq \Delta_{k_7}.$$

But since, for $k \geq k_5$, we have $\Delta_k \leq \Delta_{k+1}$, and all the steps are acceptable, then

$$\|R_{k_7+1}^{-T} h_{k_7+1}\| \leq \Delta_{k_7} \leq \Delta_{k_7+1},$$

and

$$\|(Z_{k_7+1}^T B_{k_7+1} Z_{k_7+1})^{-1}(Z_{k_7+1}^T \nabla l_{k_7+1} + Z_{k_7+1}^T B_{k_7+1} s_{k_7+1}^n)\| \leq \Delta_{k_7} \leq \Delta_{k_7+1}.$$

The above two inequalities and the fact that for all $k \geq k_5$ all the steps are acceptable imply that the full step will be taken at iteration $k_7 + 1$. By induction, for all $k \geq k_7$, the trust region will be inactive and the full step will be accepted.

This means that the sequence $x_k, k \geq k_7$ generated by the algorithm is the sequence of iterates generated by Algorithm 1.1 and consequently the local rate of convergence is q-quadratic. □

THEOREM 6.12. *Under the global and the local assumptions, if an approximation to the Hessian of the Lagrangian that satisfies (6.8) is used, then for k sufficiently large, $x_k \longrightarrow x_\star$ q-superlinearly.*

Proof. From the above theorem, we have for all $k \geq k_7$, the trust region will be inactive and the full step will be accepted, where k_7 is some sufficiently large integer. This means that the sequence $\{x_k\}, k \geq k_7$ generated by the algorithm is purely the sequence of iterates that is generated by Algorithm 1.1.

Second, it is proved by Boggs, Tolle and Wang (1982)[1] that if we use a scheme for approximating B_k in Algorithm 1.1, then $x_k \longrightarrow x_\star$ q-superlinear if and only if assumption (6.1) is satisfied.

Now as a consequence of the local assumptions and the above two parts of the proof, if k_8 is taken sufficiently large such that the local method, *i.e.* Algorithm 1.1, generates steps that are q-superlinear, we conclude that the local rate of convergence is q-superlinear. □

7. Concluding Remarks. We have presented an algorithm for solving the equality constrained optimization problem. This algorithm has many desirable features. In this algorithm, we use Fletcher's differentiable penalty function as a merit function.

In computing the trial step, after factorizing ∇h_k using QR factorization, two unexpensive subproblems has to be solved. One of them is an upper triangular linear system. The second one is a subproblem of smaller dimension $m \times m$ similar to the one we obtain when solving unconstrained optimization problems using a trust-region method.

In our algorithm, to obtain the matrix B_k, the exact Hessian of the Lagrangian can be used. On the other hand, an approximation to the Hessian matrix can also be used. For example, setting B_k to a fixed matrix for all k is valid. However, if B_k is obtained by quasi–Newton updates, the uniform boundedness assumption on B_k, condition (4.1), causes some difficulties. For an analysis of this problem for trust–region algorithms for unconstrained problems see *e. g.* Powell (1984)[18], and for minimization problems with convex constraints, see *e.g.* Toint (1988)[22]. The question of how to use a secant approximation to the Hessian of the Lagrangian is a research topic. We believe that Tapia (1988)[20] will be of considerable value here.

One of the main advantages of this algorithm is the way of updating the penalty parameter. It is updated in such way to ensure that the merit function is decreased at each iteration by at least a fraction of Cauchy decrease in the quadratic model of the linearized constraints and at the same time can be decreased whenever it is warranted.

We have presented a convergence theory for this algorithm. We showed that the algorithm is well defined and is globally convergent. To the best of our knowledge this is the first time a global convergence theory is proved for an algorithm with a non-monotonic penalty parameter updating scheme. This updating scheme should avoid the numerical difficulties that may occur if the penalty parameter is increased at each iteration. We have also proved that, the algorithm will terminate at a point that is not bounded away from a stationary point.

We also presented a local analysis for this algorithm. In our local analysis we proved that our globalization strategy will not disrupt the fast local rate of convergence.

Acknowledgment

This work was done while the author was visiting the Department of Computational and Applied Mathematics and the Center of Research on Parallel Computations, Rice University. He thanks Rice University for its financial support and for the congenial scientific atmosphere that it provided.

The author is also greatly indebted to Richard Byrd, John Dennis, and Richard Tapia for their valuable suggestions and comments on an earlier version of this paper.

REFERENCES

[1] P. T. BOGGS, J. W. TOLLE, and P. WANG. On the local convergence of quasi-Newton methods for constrained optimization. *SIAM J. Control Optim.*, 20:161–171, 1982.

[2] R. H. BYRD, R. B. SCHNABEL, and G. A. SHULTZ. A trust-region algorithm for nonlinearly constrained optimization. *SIAM J. Numer. Anal.*, 24:1152–1170, 1987. Also available as Technical Report CU-CS-313-85, Department of Computer Science, University of Colorado, Boulder, CO.

[3] T. R. COLEMAN and A. R. CONN. Nonlinear programming via an exact penalty function: asymptotic analysis. *Math. Prog.*, 24:123–136, 1982.

[4] J. E. DENNIS Jr. and R. B. SCHNABEL. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Englewood Cliffs, NJ, 1983. Russian edition, Mir Publishing Office, Moscow, 1988, O. Burdakov, translator.

[5] M. M. EL-ALEM. *A global convergence theory for a class of trust-region algorithms for constrained optimization.* PhD thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1988.

[6] M. M. EL-ALEM. A global convergence theory for the Celis-Dennis- Tapia trust-region algorithm for constrained optimization. *SIAM J. Numer. Anal.*, 28:266–290, 1991.

[7] R. FLETCHER. Second order correction for nondifferentiable optimization. In G.A.Watson, editor, *Lectures Notes in Mathematics 912.* Springer-Verlag, Berlin-New York, 1982.

[8] P. GILL, W. MURRAY, M. SAUNDERS, and M. WRIGHT. Some theoretical properties of an augmented Lagrangian merit function. Report SOL 86-6R, Department of Operations Research, Stanford University, Stanford, CA 94305-4022, 1986.

[9] P. E. GILL and W. MURRAY. Newton-type method for linearly constrained optimization. In P. E. Gill and W. Murray, editors, *Numerical methods for constrained optimization.*, pages 29–92. Academic Press, London, New York, and San Francisco, 1974.

[10] J. GOODMAN. Newton's method for constrained optimization. *Math. Prog.*, 33:162–171, 1985.

[11] M. C. MACIEL. *A global convergence theory for a general class of trust-region algorithms for equality constrained optimization.* PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1992.

[12] N. MARATOS. *Exact penalty function algorithms for finite dimensional and control optimization problems.* PhD thesis, University of London, London, 1978.

[13] D. Q. MAYNE and E. POLAK. A superlinearly convergent algorithm for constrained optimization problems. *Math. Prog. Stud.*, 16:45–61, 1982.

[14] J. NOCEDAL and M. OVERTON. Projected Hessian updating algorithms for nonlinearly constrained optimization. *SIAM J. Numer. Anal.*, 22:821–850, 1985.

[15] E. O. OMOJOKUN. *Trust-region strategies for optimization with nonlinear equality and inequality constraints.* PhD thesis, Department of Computer Science, University of Colorado, Boulder, Co, 1989.

[16] M. J. D. POWELL. Convergence properties of a class of minimization algorithms. In O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, editors, *Nonlinear Programming 2*, pages 1–27. Academic Press, New York, NY, 1975.

[17] M. J. D. POWELL. Variable metric methods for constrained optimization. In M. Grotschel A. Bachem and B. Korte, editors, *Mathematical Programming: The state of the art.*, pages 288–311. Springer-Verlag, Barlin and New York, 1983.

[18] M. J. D. POWELL. On the global convergence of trust-region algorithms for unconstrained optimization. *Math. Programming*, 29:297–303, 1984.

[19] M. J. D. POWELL and Y. YUAN. A trust-region algorithm for equality constrained optimization. *Math. Prog*, 49:189–211, 1991.

[20] R. TAPIA. On secant update for use in general constrained optimization. *Math. Comp.*, 51:181–202, 1988.

[21] R. A. TAPIA. Quasi-Newton methods for equality constrained optimization: equivalence of existing methods and a new implementation. In O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, editors, *Nonlinear Programming 3*, pages 125–164. Academic Press, New York, NY, 1978.

[22] Ph. TOINT. Global convergence of a class of trust-region methods for non–convex minimization in Helbert spaces. *IMA Journal of Numerical Analysis*, 8:231–252, 1988.

[23] K. A. WILLIAMSON. *A robust trust-region algorithm for nonlinear programming*. PhD thesis, Department of Mathematical Sciences, Rice University, Houston, Tx 77251-1892, 1990.

[24] J. Z. ZHANG and D. T. ZHU. Projected quasi-Newton algorithm with trust region for constrained optimization. *J. Opt. Th. and Appl.*, 67:369–393, 1990.