A Global Convergence Theory for a Class of Trust Region Algorithms for Constrained Optimization[1,2]

by

Mahmoud El Alem[2]

Technical Report 88-5, May 1988

---

[1]A Thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, Rice University.

[2]Research sponsored by SDIO/IST/ARO, AFOSR 85-0243, and DOE DEFG05-86ER 25017.

[2]The author received support from Alexandria University, Alexandria, Egypt.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
MAY 1988		**00-00-1988 to 00-00-1988**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Global Convergence Theory for a Class of Trust Region Algorithms for Constrained Optimization	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Computational and Applied Mathematics Department ,Rice University,6100 Main Street MS 134,Houston,TX,77005-1892**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**127**	

RICE UNIVERSITY

A GLOBAL CONVERGENCE THEORY FOR A CLASS OF TRUST REGION ALGORITHMS FOR CONSTRAINED OPTIMIZATION

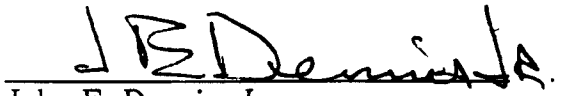
by

Mahmoud Mahmoud El-Alem

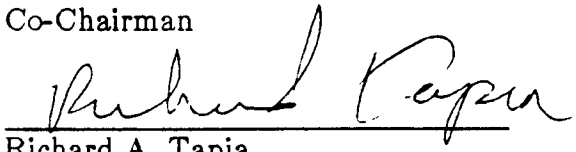
A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

DOCTOR OF PHILOSOPHY

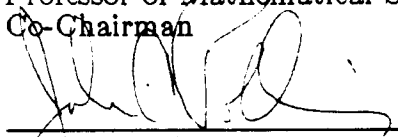
Approved, Thesis Committee:



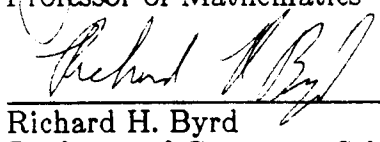
John E. Dennis, Jr.
Professor of Mathematical Sciences
Co-Chairman



Richard A. Tapia
Professor of Mathematical Sciences
Co-Chairman



John C. Polking
Professor of Mathematics



Richard H. Byrd
Professor of Computer Sciences
University of Colorado

Houston, Texas

May, 1988

RICE UNIVERSITY

A GLOBAL CONVERGENCE THEORY FOR A CLASS OF TRUST REGION ALGORITHMS FOR CONSTRAINED OPTIMIZATION

by

MAHMOUD MAHMOUD EL-ALEM

DOCTOR OF PHILOSOPHY

A GLOBAL CONVERGENCE THEORY FOR A CLASS OF TRUST REGION ALGORITHMS FOR CONSTRAINED OPTIMIZATION

MAHMOUD MAHMOUD EL-ALEM

ABSTRACT

In this research we present a trust region algorithm for solving the equality constrained optimization problem. This algorithm is a variant of the 1984 Celis-Dennis-Tapia algorithm. The augmented Lagrangian function is used as a merit function. A scheme for updating the penalty parameter is presented. The behavior of the penalty parameter is discussed.

We present a global and local convergence analysis for this algorithm. We also show that under mild assumptions, in a neighborhood of the minimizer, the algorithm will reduce to the standard SQP algorithm; hence the local rate of convergence of SQP is maintained.

Our global convergence theory is sufficiently general that it holds for any algorithm that generates steps that give at least a fraction of Cauchy decrease in the quadratic model of the constraints.

In the name of Allah, the Beneficent, the Merciful.

"Allah will exalt those who believe among you,
and those who have knowledge,
to high ranks"
The Holy Qur'an, Sura 58, Ayah 11.

"Seeking knowledge is obligatory on every muslim"
Prophet Muhammad (peace be upon him).

"Whoever goes in a way seeking knowledge
Allah will facilitate with it an easy way to Paradise"
Prophet Muhammad (peace be upon him).

TO MY MOTHER

AND

THE MEMORY OF MY FATHER

ACKNOWLEDGEMENTS

I would like to express my deepest thanks to my advisors, Professor John Dennis and Richard Tapia, for their support, guidance and encouragement. I wish also to thank Mrs. Ann Dennis and Mrs. Jean Tapia for their kindness and help. I am deeply grateful to Professor Richard Byrd for his friendship, encouragement, guidance and valuable discussions throughout this research.

My deepest appreciation goes to my mother who sacrificed much in order for me to pursue my graduate studies.

I owe a great deal of thanks to Professor David Sanchez Professor of Mathematics and Provost, Lehigh University for helping me in my first semester at University of New Mexico.

I have deeply appreciated the encouragement of Professor Francesco Zirilli, University De Rome, and of Dr. Mohammedi El-Hallabi, Ecole Hassainq des Tavaux, Morocco.

Mathematical and personal thanks go to Professor Mahmoud El-Borai, Chairman of the Mathematics Department, Alexandria University, Egypt, to Amr El-Bakry of Rice University and to Emmanuel Omojokun of University of Colorado.

For always being very helpful I thank Omar (Douglas) Monk and Basalat Ali Raja of Rice University.

I would like to express my gratitude to Alexandria University, Egypt for giving me the opportunity to complete my study at Rice University, and to the Egyptian Cultural and Educational Bureau for supporting my graduate work in the U.S.

For keeping the department running and looking after the graduate program, I thank Judy Forney.

Table of Contents

CHAPTER ONE

INTRODUCTION

This chapter consists of two parts. In the first part we define the general optimization problem and some special cases of this problem. We also state the optimality conditions for some of these special cases. The second part is devoted to presenting from the historical point of view, some methods that attempt to solve the equality constrained optimization problem.

1.1 CLASSIFICATION OF THE PROBLEMS

By the general optimization problem we mean the problem of finding $x_* \in S$ that solves the following problem:

$$\begin{aligned} &\text{minimize } f(x), \\ &\text{subject to } x \in S, \end{aligned} \tag{GOP}$$

where f is assumed to be a smooth nonlinear function defined from S into R. A point $x_* \in S$ is said to be a local solution of problem (GOP) if there exists a neighborhood $N(x_*)$ such that $f(x_*) \leq f(x)$ for all $x \in N(x_*) \bigcap S$.

The optimization problem can be characterized by the type of set S on which f is to be minimized. If S is R^n, then the problem will be referred to as an unconstrained optimization problem or problem (UCOP). It can be written as:

$$\underset{x \in R^n}{\text{minimize}} \quad f(x)\,, \tag{UCOP}$$

i.e., the unconstrained optimization problem is the problem of minimizing f without constraints.

If $f \in C^1$, then a necessary condition for $x_* \in R^n$ to be a solution of (UCOP) is

$$\nabla f(x_*) = 0\,,$$

where ∇f denotes the gradient of f. It is also necessary, if $f \in C^2$, that the Hessian of f at x_* be positive semidefinite.

Sufficient conditions for $x_* \in R^n$ to be a local solution of (UCOP) are:

$$\begin{aligned} \nabla f(x_*) &= 0\,, \\ v^T \nabla^2 f(x_*)\, v &> 0\,, \end{aligned}$$

for all nonzero vectors $v \in R^n$.

On the other hand, if S can be defined by a set of equality and inequality constraints then the problem will be called the general nonlinear programming problem. So, by the general nonlinear programming problem we mean the constrained optimization problem:

$$\begin{aligned} &\text{minimize} && f(x)\,, \\ &\text{subject to} && h_i(x) = 0 \qquad i=1,\ldots,m, \\ &&& g_j(x) \geq 0 \qquad j=1,\ldots,p\,. \end{aligned} \tag{NLP}$$

where f, h_i, and g_j are assumed to be smooth nonlinear functions defined from R^n into R.

As a special case of this, if we seek to minimize f on a manifold S defined by equations of the form:

$$h_i(x) = 0 \qquad i=1,\ldots,m < n\,,$$

i.e., we are concerned with the case where only equality constraints are involved, then we refer to this problem as the equality constrained optimization problem or problem (EQ), and it can be expressed as:

$$\begin{aligned} &\text{minimize } f(x), \\ &\text{subject to } h_i(x) = 0 \qquad i=1,\ldots,m . \end{aligned} \tag{EQ}$$

On the other hand, we will refer to the problem in which only inequality constraints are involved as the inequality constrained optimization problem or problem (INEQ). It can be expressed as:

$$\begin{aligned} &\text{minimize } f(x), \\ &\text{subject to } g_i(x) \geq 0 \qquad i=1,\ldots,p . \end{aligned} \tag{INEQ}$$

In this research, we consider only the equality constrained optimization or problem (EQ). We will denote by $h(x)$ the vector whose components are $h_i(x)$ $i=1,\ldots,m$. When f and $h \in C^2$, we will say problem (EQ) $\in C^2$. It is convenient to introduce the Lagrangian function $l : R^n \times R^m \rightarrow R$ associated with problem (EQ). It is the function:

$$l(x, \lambda) = f(x) + \lambda^T h(x), \tag{1.1.1}$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)^T$ is called the Lagrange multiplier.

Stating necessary optimality conditions in terms of the Lagrangian function requires a constraint qualification. A satisfactory but somewhat restrictive constraint qualification is the regularity assumption: that is, the vectors $\nabla h_i(x)$ $i=1,\ldots,m$ are linearly independent. Any feasible point at which the regularity assumption is satisfied is called a regular point. We will use the notation $\nabla h(x)$ to mean the matrix whose columns are $\nabla h_i(x)$ $i=1,\ldots,m$.

The first order necessary conditions, or Kuhn-Tucker conditions, are that x_* be a

feasible point (*i.e.* $h\ (x_*) = 0$), and that there exists a Lagrange multiplier λ_* such that:

$$\nabla l(x_*,\lambda_*) = 0\ .$$

The second order necessary condition is that the Hessian of the Lagrangian function is positive semidefinite for all vectors that lie in the null space of $\nabla h(x_*)^T$. That is, for all v that satisfies: $\nabla h(x_*)^T\ v = 0$, we have

$$v^T\ \nabla_x^2 l(x_*\ ,\ \lambda_*)\ v \geq 0\ .$$

Sufficient conditions for x_* to be an isolated local minimizer of problem (EQ) are that x_* is a Kuhn-Tucker point (*i.e.* x_* and λ_* satisfy the first order necessary conditions), and that

$$v^T\ \nabla_x^2 l(\ x_*\ ,\ \lambda_*\ )\ v\ > 0$$

for every nonzero vector v that satisfies $\nabla h(x_*)^T\ v = 0$.

1.2 HISTORICAL BACKGROUND

In this section we present some methods that attempt to solve problem (EQ). We start with the sequential unconstrained minimization techniques, or (SUMT), which were popularized by A. Fiacco and G. McCormick in the late 60's. Then we present the multiplier methods which were famous in the early 70's. After that we present five different ways to extend Newton's method from unconstrained optimization to constrained optimization.

1.2.1) The Penalty Function Methods

Some of the earliest practical approaches for solving problem (EQ) were the sequential unconstrained minimization techniques or (SUMT). These techniques are based on solving a sequence of unconstrained minimization problems whose solutions approach the solution of problem (EQ). Penalty function methods belong to this class. [See Fiacco and McCormick (1968)]

Penalty function methods solve a sequence of minimization subproblems in which a "penalty" term for constrained violation is added to the objective function. The first penalty function was suggested by Courant (1943) for problem (EQ). It is the function:

$$P(x,r) = f(x) + \frac{1}{2}\, r\, h(x)^T h(x) \qquad\qquad r \geq 0 .$$

It can be shown under mild assumptions that if $x(r)$ is a minimizer of $P(x,r)$ for every r, then:

$$\lim_{r \to \infty} x(r) = x_* ,$$

where x_* is a solution to problem (EQ) [see for example Poljak(1971)]. The penalty function methods can be stated as follows:

ALGORITHM (1.2.1): Penalty Function Method

1) Given x_0, choose $r_0 \geq 0$.

2) For $k = 1,2,...$ until convergence do

i) Find x_{k+1} such that:

$$x_{k+1} = \operatorname*{argmin}_x P(x,r_k) .$$

ii) Choose $r_{k+1} \geq r_k$.

These methods generate a sequence of infeasible points. In fact, each iterate is either necessarily infeasible or a solution of problem (EQ). These methods are not appropriate for problems in which feasibility must be maintained.

The availability of powerful methods for solving unconstrained optimization problems, the well-developed theoretical background and the comparative simplicity of these methods are attractive. However, in practice it is inefficient to require that the sequence of unconstrained minimization problems be solved exactly and they suffer from severe numerical difficulties since the unconstrained problems that must be solved become increasingly more ill-conditioned as the solution is approached.

1.2.2) The Multiplier Method

It is well known that in order to guarantee convergence of the penalty function methods the penalty parameter must go to infinity, and so the problem becomes increasingly ill-conditioned. Therefore, it would be useful to derive methods for which the parameters need only assume moderate values.

These concerns motivated Hestenes (1969) to introduce his multiplier method. He suggested the augmented Lagrangian function:

$$L(x,\lambda;r) = f(x) + \lambda^T h(x) + r\ h(x)^T h(x)\ . \qquad (1.2.1)$$

The multiplier method consists of updating an estimate of the Lagrange multiplier λ and sometimes the penalty parameter at each iteration. The multiplier method can be stated as follows:

ALGORITHM (1.2.2): The Multiplier Method

1) Given $x_0 \in R^n$ and $r_0>0$, determine $\lambda_0 \in R^m$.

2) For $k = 1,2,...$ until convergence do

i) Find x_{k+1} such that:

$$x_{k+1} = \underset{x}{\operatorname{argmin}} \quad L(x,\lambda_k,r_k) .$$

ii) Update r_k by some update formula.

iii) Update λ_k by some multiplier update formula.

As an update formula for the estimate of the multiplier, Hestenes (1969) and independently Powell (1969) suggested:

$$\lambda_{k+1} = \lambda_k + r_k \, h(x_k) . \tag{1.2.2}$$

Haarhoff and Buys (1970) proposed:

$$\lambda_{k+1} = -(\nabla h_{k+1}^T \nabla h_{k+1})^{-1} \nabla h_{k+1}^T \nabla f_{k+1} . \tag{1.2.3}$$

Buys (1972) proposed:

$$\lambda_{k+1} = \lambda_k + (\nabla h_k^T \nabla_x^2 L_k^{-1} \nabla h_k)^{-1} h_k . \tag{1.2.4}$$

Miele (1972) proposed:

$$\lambda_{k+1} = (\nabla h_{k+1}^T \nabla h_{k+1})^{-1} (h_{k+1} - \nabla h_{k+1}^T \nabla f_{k+1}) . \tag{1.2.5}$$

The formula:

$$\lambda_{k+1} = (\nabla h^T \nabla_x^2 L_k^{-1} \nabla h_k)^{-1} \, [\, h_k - \nabla h^T \nabla_x^2 L_k^{-1} \, (\nabla f_k + r_k \nabla h_k h_k) \,] . \tag{1.2.6}$$

was suggested by Tapia (1974a), (1974b) in a different context.

A complete analysis of the multiplier methods was presented by Tapia

(1977). Computational experience with the multiplier methods was reported by Miele *et al.* (1971a), (1971b), (1972a), and (1972b).

It is shown by Buys (1972) that if we define the dual of problem (EQ) to be the following problem:

$$\max_{\lambda} \min_{x} L (x , \lambda ; r) ,$$

where L is the augmented Lagrangian function (1.2.1) and r is a sufficiently large fixed penalty parameter, then if x_* solves the primal problem (*i.e.* x_* solves problem (EQ)), then its associated Lagrange multiplier λ_* solves the dual problem and x_* can be obtained from λ_* as the solution of

$$\min_{x} L (x , \lambda_* ; r) .$$

The multiplier method with multiplier update formula (1.2.2) or (1.2.3) is the gradient method applied to the dual problem; and the multiplier method with multiplier update formula (1.2.4) or (1.2.6) is Newton's method applied to the dual problem. [Buys (1972)]

Based on these facts, the rate of convergence of the multiplier method with a sufficiently large fixed penalty parameter using (1.2.2) or (1.2.3) as an update formula for the multiplier can be shown to be q-linear in x and in λ. Additional results show that it is q-superlinear in λ if and only if the penalty parameter goes to infinity [see for example Bertsekas (1976)]. On the other hand, the rate of convergence of the multiplier methods with a sufficiently large fixed penalty constant using (1.2.4) or (1.2.6) as an update formula for the multiplier can be shown to be q-quadratic in x and in λ.

1.2.3) Newton's Method For Problem (EQ)

Up to this point, from the historical and chronological points of view we have seen that the price we pay for convergence in the penalty function methods is a deterioration in numerical conditioning, since the penalty parameter must go to infinity. The parameterized subproblem that has to be solved at each iteration in the multiplier method suffers from ill-conditioning since the penalty parameter has to be set to a sufficiently large value. In the multiplier method using (1.2.2) or (1.2.3) as an update formula for the estimate of the multiplier, in order to guarantee fast convergence, again the penalty parameter must go to infinity, and the problem becomes increasingly ill-conditioned. Although the multiplier method using (1.2.4) or (1.2.6) gives fast local convergence, it still suffers from the fact that the subproblem requires a complete minimization in x in order to get a step. To address these problems, an algorithm which would give fast convergence without a corresponding deterioration in numerical conditioning is needed. Such an algorithm is presented in this section.

Five different ways to extend Newton's method from unconstrained optimization to constrained optimization have been suggested. These are the extended problem, the successive quadratic programming, the diagonalized multiplier method, the structured multiplier substitution method, and Goodman's method.

This section is devoted to a discussion of each of these ways. We start our discussion of extending Newton's method to problem (EQ) by considering the extended problem. Then we will consider the successive quadratic programming, the diagonalized multiplier method, the structured multiplier substitution method, and finally Goodman's method.

1.2.3.1) The Extended Problem:

Suppose problem (EQ) $\epsilon\ C^2$. Let x_* be a local solution which is also a regular point. The first order necessary conditions and the regularity assumption imply that there exists a Lagrange multiplier λ_* such that (x_*,λ_*) is a solution of the following nonlinear system:

$$\begin{aligned} \nabla_x l(x,\lambda) &= 0 \\ h(x) &= 0. \end{aligned} \tag{1.2.2}$$

Following Tapia (1977), (1978), by the extended system we mean the nonlinear system of equations (1.2.2), and by the extended problem corresponding to problem (EQ) we mean the problem of finding a stationary point of the Lagrangian function. *i.e.* solving for a root of the extended system. Now, consider applying Newton's method to solve the extended problem. Our assumption will be the standard assumptions of Newton's method. Specifically, we assume the following:

(1) f and $h\ \epsilon\ C^2$.

(2) $\nabla^2 l(x_*,\lambda_*)$ is invertible

(3) $\nabla_x^2 l$ is Lipschitz continuous with respect to x in a neighborhood of the solution.

Newton's method on the extended system can be stated as follows:

ALGORITHM (1.2.3) Newton's Method on the Extended System

1) Given $x_0\ \epsilon\ R^n$ and $\lambda_0\ \epsilon\ R^m$

2) For $k = 1,2,...$ until convergence do

i) Solve for $(\ s\ ,\ \Delta\lambda\ )$ the following linear system

$$\nabla_x^2 l_k s + \nabla h_k \Delta\lambda = -\nabla_x l_k$$
$$\nabla h_k^T s = -h_k \ .$$

ii) Set: $x_{k+1} = x_k + s$.

iii) Set: $\lambda_{k+1} = \lambda_k + \Delta\lambda$.

Under the standard assumptions of Newton's method, this algorithm gives local q-quadratic convergence in (x,λ). [See Tapia (1977)]

1.2.3.2) The Successive Quadratic Programming Method (SQP):

The successive quadratic programming method is effective for solving problem (EQ). Algorithms of this type compute the minimizer of problem (EQ) by solving a sequence of quadratic programming subproblems. Namely, by the successive quadratic programming method or (SQP), we mean the iterative procedure:

ALGORITHM (1.2.4): The Successive Quadratic Programming Method

1) Given $x_0 \in R^n$, determine $\lambda_0 \in R^m$.

2) For $k = 1,2,...$ until convergence do

i) Find a solution $(s^{QP}, \Delta\lambda^{QP})$ to the following quadratic programming problem:

$$\text{minimize} \quad \nabla_x l_k^T s + \frac{1}{2} s^T \nabla_x^2 l_k \, s$$
$$\text{subject to} \quad h_k + \nabla h_k^T s = 0 \ .$$

ii) Set $x_{k+1} = x_k + s^{QP}$.

iii) Set $\lambda_{k+1} = \lambda_k + \Delta\lambda^{QP}$.

1.2.3.3) The Diagonalized Multiplier Method (DMM):

At each iteration, the multiplier method described in Section (1.2.3) goes through a complete minimization step for x and only one update for λ, although we are solving for both the minimizer x_* and its associated multiplier λ_*. It would then make sense to update the estimate of the multiplier after each update of the minimizer. This idea motivated Tapia (1977) to introduce the diagonalized multiplier method. It can be written as follows:

ALGORITHM (1.2.5): The Diagonalized Multiplier Method

1) Given $x_0 \in R^n$, determine $\lambda_0 \in R^m$.

2) For $k = 1,2,...$ until convergence do

   i) Update λ_k by some multiplier update formula.

   ii) Calculate:

$$x_{k+1} = x_k - \nabla_x^2 l_k^{-1} \nabla_x l(x_k, \lambda_{k+1}) .$$

1.2.3.4) Structured Multiplier Substitution Method (SMSM):

Consider an estimate of the Lagrange multiplier of the form:

$$\lambda(x) = (\nabla h(x)^T D \nabla h(x))^{-1} (h(x) - \nabla h(x)^T D \nabla f(x)) , \qquad (1.2.3)$$

where D is any $n \times n$ positive semi-definite matrix that may depend on x. Then $\nabla L(x, \lambda(x)) = 0$ is equivalent to $(x , \lambda(x))$ being a stationary point of the augmented Lagrangian given by (1.2.1). This powerful fact motivated Tapia (1978) to introduce the multiplier substitution method. The idea is straightforward; solve for a root of the following problem:

$$\nabla_x L(x,\lambda(x)) = 0 . \qquad (1.2.4)$$

using any iterative scheme.

By the multiplier substitution Newton's method, we mean the multiplier substitution method using Newton's method as an iterative scheme to solve (1.2.4).

By the structured multiplier substitution method, we mean the multiplier substitution method taking the advantage of omitting the terms that vanish at the solution. This method can be stated as an algorithm as follows:

ALGORITHM (1.2.6): Structured Multiplier Substitution Method

1) Given $x_0 \in R^n$, determine $\lambda_0 \in R^m$.

2) For $k = 1,2,...$ until convergence do

i) Solve for s the following linear system

$$(I - A\ D)\ [\ \nabla_x^2 L_k s + \nabla f_k\ ] + \nabla h_k\ (\nabla h_k^T D\, \nabla h_k)^{-1} [\ \nabla h_k^T s + h_k\ ] = 0 ,$$

where $A(x) = \nabla h(x)(\nabla h(x)^T D\, \nabla h(x))^{-1} \nabla h(x)^T$.

ii) Set $x_{k+1} = x_k + s$

The four methods discussed in this section are equivalent. Specifically, Tapia (1978) showed that for problem (EQ), the extended problem with the Lagrangian function given by (1.1.1), the successive quadratic programming method, the diagonalized multiplier method, and the structured multiplier substitution method generate identical (x,λ) iterates.

1.2.3.5) Goodman's Method (GM):

Let $x_* \in R^n$ be a feasible point. If $z_1(x_*), \ldots, z_{n-m}(x_*)$ are a basis for the null space of $\nabla h(x_*)^T$, then a necessary condition for x_* to be a local minimizer of problem (EQ) is

$$\nabla f(x_*)^T \, z_i(x_*) = 0, \qquad i = 1,\ldots,n-m . \tag{1.2.5}$$

If we define $z_i(x)$ in a neighborhood of x_*, and let $Z(x)$ be the matrix whose columns are $z_i(x)$, $i = 1,\ldots,n-m$, then Goodman's method can be defined to be the method that uses Newton's method to solve the following $n \times n$ nonlinear system

$$\begin{aligned} Z(x)^T \nabla f(x) &= 0 \\ h(x) &= 0 . \end{aligned}$$

This method can be stated as follows:

ALGORITHM (1.2.7): Goodman's Method

1) Given $x_0 \in R^n$, determine $\lambda_0 \in R^m$.

2) For $k = 1,2,\ldots$ until convergence do

i) Form a basis $Z(x_k)$ for the null space of $\nabla h(x_k)^T$.

ii) Find a solution s to the following linear system:

$$\begin{aligned} Z(x_k)^T W(x_k) \, s &= - Z(x_k)^T \nabla f(x_k) \\ \nabla h(x_k)^T \, s &= - h(x_k) . \end{aligned}$$

where $W(x_k)$ is the Hessian of the Lagrangian function

$f(x) + h(x)^T \lambda(x_k)$.

iii) Set $x_{k+1} = x_k + s$.

It is easy to see that for problem (EQ), Goodman's method is equivalent to the successive quadratic programming method using the projection formula (1.2.3) to update the estimate of the Lagrange multiplier. [Goodman (1985)]

Of these equivalent formulations, the SQP method is the most visible and popular. The main reason for its popularity is that it allows inclusion of inequality constraints in a straightforward manner. To do so, one merely carries them along as linearized inequalities in the quadratic program. Another reason for its popularity is that the SQP approach allows use of existing quadratic programming modules in its implementation.

From a theoretical point of view, the extended problem plays a very important role and has been in the background of the derivation of many algorithms. This formulation is widely used for the convergence analysis of its equivalent methods.

CHAPTER TWO

GLOBALIZATION STRATEGIES

It is known that Newton's method is locally q-quadratically convergent under reasonable hypothesis. This means that there exists a neighborhood of the solution such that if the starting point lies in that neighborhood, the sequence of iterates generated by the method will converge rapidly to that solution.

This chapter deals with modifications to such methods that attempt to force convergence to a solution from any starting point without sacrificing fast local convergence.

This chapter consists of two parts. In the first part we discuss the globalization strategy for Newton's method by considering the unconstrained optimization problem. In Sections 2.1.1 and 2.1.2 we discuss in some detail the two main globalization strategies. The second part is devoted to study in detail the globalization strategy for problem (EQ). A crucial ingredient is the use of a merit function. In Section 2.2.1 we discuss some of the existing merit functions. In Section 2.2.2 we present some existing methods for solving problem (EQ).

2.1 GLOBALIZATION STRATEGY FOR PROBLEM (UCOP)

We start our discussion of globalizing Newton's method by considering the unconstrained optimization problem or problem (UCOP). In this section we discuss the two main globalization strategies: namely, the line search strategy and

the model trust region strategy.

2.1.1) Line Search Globalization Strategy

This is the modern version of the traditional idea of backtracking along Newton's direction if a full Newton's step is unsatisfactory.

The idea of the line search strategy is simple and natural. Let s_k be Newton's step at x_k. We take a step $\gamma_k s_k$, for some $\gamma_k > 0$, that makes $x_{k+1} = x_k + \gamma_k s_k$ an acceptable next iterate.

An acceptable step at least has to satisfy the so called α-condition

$$f(x_k + \gamma_k\ s_k) \leq f(x_k) + \gamma_k \alpha \nabla f(x_k)^T s_k\ ,$$

where $\alpha \in (0,1)$ is a small fixed constant. An additional condition may also be required. Different rules may be used to define an acceptable step. Some of these rules were studied by Armijo (1969), Goldstein (1967) and Wolfe (1969).

The convergence theory of such an algorithm shows that choosing $\gamma_k = 1$ whenever it is acceptable will not affect the fast local convergence [see Dennis and Moré (1977)]. This fact suggested an algorithm for choosing γ_k. The idea is simple, we start with $\gamma_k = 1$, and then, if $x_k + s_k$ is not acceptable, backtrack by decreasing γ_k until an acceptable $x_k + \gamma_k s_k$ is found. This is precisely the backtracking algorithm.

ALGORITHM (2.1.1): The Backtracking Algorithm

Given $\alpha \in (0,1)$, $0 < l < u < 1$ and $\gamma_k = 1$

while $f(x_k + \gamma_k\ s_k) > f(x_k) + \gamma_k\ \alpha\ \nabla f(x_k)\ s_k$

do

$\gamma_k := \rho\, \gamma_k$ for some $\rho \in [l\ ,\ u]$

$x_{k+1} := x_k + \gamma_k\, s_k$.

For more details concerning line-search strategies we refer the reader to Dennis and Schnabel (1983).

2.1.2) Trust Region Globalization Strategy

The idea of the trust region is based on estimating the region in which a local model of the function f at x_k can be trusted to adequately represent the function, and then taking the step which minimizes the model in this region.

Specifically, we build a local model of $f(x_k+s_k)$ at x_k , say $m_k(s_k)$, which at least satisfies the properties:

$$m_k(\,0\,) = f(\,x_k\,)\,, \tag{2.1.1}$$

$$\nabla m_k(\,0\,) = \nabla f(\,x_k\,)\,. \tag{2.1.2}$$

Given such a model and a trust region radius Δ_k , we solve for s_k the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad m_k\,(\,s\,) \\ &\text{subject to} \quad ||\,s\,||_2 \leq \Delta_k\ . \end{aligned}$$

If the model $m_k(s_k)$ is good enough, *i.e.*, if

$$\frac{Ared_k}{Pred_k} \geq \eta_1\,, \tag{2.1.3}$$

where $\eta_1 \in (0,1)$ is a small fixed constant,

$$Ared_k = f(x_k) - f(x_k+s_k), \tag{2.1.4}$$

and

$$Pred_k = f(x_k) - m_k(s_k), \tag{2.1.5}$$

then we accept the step s_k and set $x_{k+1} = x_k + s_k$.

If the local model $m_k(s_k)$ is convex, we obtain:

$$\nabla m_k(0)^T s_k \leq m_k(s_k) - m_k(0).$$

This relation, using inequalities (2.1.1) and (2.1.2), can be written as:

$$\nabla f(x_k)^T s_k \leq m_k(s_k) - f(x_k). \tag{2.1.6}$$

Using (2.1.4) and (2.1.5), we can rewrite (2.1.3) as:

$$f(x_k+s_k) \leq f(x_k) + \eta_1 (m_k(s_k) - f_k(0)).$$

which, because of (2.1.6), can be viewed as a relaxation of the α-condition:

$$f(x_k+s_k) \leq f(x_k) + \alpha \nabla f(x_k)^T s_k \tag{2.1.7}$$

As a criterion used to accept or reject the step s_k, Moré and Sorensen (1983) use (2.1.3) and Dennis and Schnabel (1983) use (2.1.7).

If the step s_k is rejected, then we set $x_{k+1} = x_k$ and decrease the radius of the trust region for the next iteration by choosing:

$$\Delta_{k+1} \in [\alpha_1 \| s_k \|_2, \alpha_2 \| s_k \|_2],$$

where $0 < \alpha_1 \leq \alpha_2 < 1$.

On the other hand, if the step is accepted, we set $x_{k+1} = x_k + s_k$ and Δ_k is updated according to the following scheme:

If

$$\frac{Ared_k}{Pred_k} < \eta_2 \quad \text{where} \quad \eta_2 \in (\eta_1, 1),$$

then the radius of the trust region is updated by setting:

$$\Delta_{k+1} = \min [\Delta_k, \alpha_3 || s_k ||_2] \quad \text{where} \quad \alpha_3 \geq 1 .$$

Else, if

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

then we update Δ_k by setting:

$$\Delta_{k+1} = \max [\Delta_k, \alpha_3 || s_k ||_2] .$$

2.2 GLOBALIZATION STRATEGY FOR PROBLEM (EQ)

Now, we consider the equality constrained optimization problem. In Section (1.2.3) we saw that the SQP algorithm is equivalent to Newton's method on the extended system. So, 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 (if we use exact second-order information). However, from the bad side of Newton's method, it is not a globally convergent method. It converges 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.

Before we start our discussion of the globalization strategy of methods that attempt to solve problem (EQ), we have to answer the following important question:

How do we test the step s_k to see if it will make satisfactory progress towards the solution of problem (EQ) in going from x_k to x_k+s_k ?

The answer to this question is not easy. It takes us to the following section.

2.2.1) Merit Functions For Constrained Optimization Problem

In the case of unconstrained optimization, it is sufficient to accept the step s_k if $f(x_k+s_k)$ is smaller than $f(x_k)$ by an appropriate amount. However, for constrained optimization, there are two goals, which may not be compatible; first, to reduce the objective function $f(x)$, and, second, to go toward feasibility.

Of course, the real problem is to identify an appropriate merit function Φ. This function should connect f and h in such a way that progress in the merit function means progress in solving the problem. There should be a connection between the merit function and the way the step is computed in the sense that the step s generated by the subproblem should give a decrease in the merit function. This decrease should be sufficient to lead to the solution of problem (EQ). It is preferred that Φ be smooth, free of arbitrary parameters, and inexpensive to evaluate. On the other hand, the merit function Φ should not disrupt the rapid rate of convergence of the basic method in a neighborhood of the solution.

We should accept the fact that no ideal merit function with all desirable properties yet exists. Some properties may only be obtained at the expense of others.

Although many merit functions have been suggested, they usually suffer from either the fact that they involve parameters for which there is no clear choice, or they are not compatible with the subproblem from which the step is computed.

Now, let us consider some merit functions that have been suggested to force global convergence.

First, consider a class of merit functions that have the following form:

$$\Phi(x) = f(x) + W(h(x)), \tag{2.2.1}$$

where $W(h(x))$ is nonnegative for all $h \in R^m$ and satisfies $W(0) = 0$. Special cases of this function are:

The least squares penalty function,

$$\Phi(x) = f(x) + r \, ||h(x)||_2^2 \tag{2.2.2}$$

is used by Bartholomew-Biggs (1982). Celis, Dennis, and Tapia (1987) used this function as a relaxed merit function.

Han (1977b) used the following l_1 penalty function:

$$\Phi(x) = f(x) + r \, ||h(x)||_1. \tag{2.2.3}$$

Many algorithms have employed such a function as a merit function [for example, see Powell (1978), Coleman and Conn (1982) and Byrd, Schnabel, and Shultz (1985)].

The function:

$$\Phi(x) = f(x) + r \, ||h(x)||_2 \tag{2.2.4}$$

is used as a merit function by Byrd, Omojokun, Schnabel, and Shultz (1987).

This class of merit functions has a very useful property that if r is any number satisfying $r > ||\lambda_*||_\infty$, then Φ has a local minimum at x_*.

The merit function of the form (2.2.2) is differentiable. However, (2.2.3) and (2.2.4) are not differentiable. A disadvantage of using a nondifferentiable merit functions is that it needs special methods to deal with the nondifferentiability and we lose the advantage of widely used, well developed algorithms that require differentiability.

All merit functions of the form (2.2.1) suffer from the Maratos effect [Maratos

(1978)] which means that:

$$\Phi (x_k + s) > \Phi (x_k). \tag{2.2.5}$$

is possible even when the trial step s makes great progress towards the solution. The following example by Maratos (1978) explains this.

Example:

Consider the following problem:

$$\begin{aligned} &\text{minimize } f(x) = -x_1 + 2 (x_1^2 + x_2^2 - 1), \\ &\text{subject to } x_1^2 + x_2^2 - 1 = 0 . \end{aligned}$$

The solution is $x_* = (1,0)^T$.

The Hessian of the Lagrangian at the solution is the unit matrix.

Now, if x_k is the point

$$x_k = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}$$

for some angle θ , then the SQP method using the unit matrix as an approximation to the Hessian at x_k , gives the following search direction:

$$s = \begin{pmatrix} \sin^2\theta \\ -\sin\theta\cos\theta \end{pmatrix},$$

and we get

$$|| x_k - x_* ||_2^2 = 2 (1 - \cos \theta),$$

$$|| x_k + s - x_* ||_2 = (1 - \cos \theta)^2 .$$

However,

$$f (x_k + s) > f (x_k)$$

$$h (x_k + s) > h (x_k).$$

So, the trial step s increases all merit functions of the form (2.2.1), even though it has the quadratic rate of convergence because

$$|| x_k+s - x_* ||_2 = \frac{1}{2} || x_k - x_* ||_2^2 .$$

This example shows that an algorithm that attempts to globalize the SQP method and employs a merit function of the form (2.2.1) may reject steps similar to the step s in the last example. Consequently, the fast local rate of convergence will be disrupted.

Another disadvantage of using a function of the form (2.2.1) as a merit function is shown in the following example [Byrd, Schnabel, and Shultz (1985)].

Consider

$$\begin{aligned} \text{minimize } & f(x) = 2\, x_1 + \frac{1}{2} x_2^2 \\ \text{subject to } & x_1^2 + x_2^2 = 1 . \end{aligned}$$

The only local minimizer is at $x_* = (-1\ ,\ 0)^T$ but there is a Kuhn-Tucker point at $x_{**} = (1\ ,\ 0)^T$ with Lagrange multiplier $\lambda_{**} = -1$.

The Hessian of the Lagrangian at that point is

$$\nabla_x^2 l(x_{**},\lambda_{**}) = \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix}$$

and

$$h(\ x_{**}\ ) = 0 .$$

Assume that the algorithm is of trust region SQP type (see Section (2.2.2) for the definition of this algorithm). Let $x_k = (1\ ,\ 0\ )^T$, then the algorithm will take a step of the form $\Delta\, s$ where Δ is the radius of the trust region and $s = (0\ ,\ 1)^T$ is in the direction of the negative gradient. However, a step of any length along the direction s will increase both the objective function and the absolute value of

the constraints. Therefore, any algorithm of that kind that passes by this point will never leave it even though it is a maximizer.

To avoid the Maratos effect, some techniques have been suggested. The first, the watchdog technique, is to relax condition (2.2.5) at some iterations [see Chamberlain, Lemarechal, Pedersen and Powell (1982)], or to add to the step what is called the second order correction [see for example Coleman and Conn (1982), Fletcher (1982), (1984), Mayne and Polak (1982), Byrd, Schnabel, and Shultz (1985)].

Adding the second order correction also takes care of the disadvantage that was described in the last example. However, it adds more expense to the trial step.

Some other useful merit functions have the following general form:

$$\Phi(x,\lambda) = f(x) + \lambda^T h(x) + W(\, h(x)\,), \tag{2.2.6}$$

where W is a continuously differentiable function that satisfies $W(\, h(x)\,)$ is nonnegative for all $h \in R^m$ and $W(0) = 0$.

One of the advantages of using a merit function of this class is that it avoids the Maratos effect that might happen if we employed one of the form (2.2.1).

One of the most natural and useful merit functions was suggested by Hestenes (1969). It is the augmented Lagrangian function:

$$\Phi(x,\lambda;r) = f(x) + \lambda^T h(x) + r \;\; || \, h(x) \, ||_2^2 \tag{2.2.7}$$

where $\lambda \in R^m$.

It is well known that

$$\Phi(x,\lambda_*;r) = f(x) + \lambda_*^T h(x) + r \;\; || \, h(x) \, ||_2^2$$

has a local minimum at x_* when r is sufficiently large, where λ_* is the

Lagrange multiplier at the solution. Since λ_* is not known except at the solution x_*, an update formula for λ must be used to approximate λ_* during the minimization calculation. In Section (1.2.2) we presented some update formulas that have been suggested. We recall three of them

i) The projection formula:

$$\lambda_{k+1} = -(\nabla h_{k+1}^T \nabla h_{k+1})^{-1} \nabla h_{k+1}^T \nabla f_{k+1} . \tag{2.2.8}$$

ii) Miele's formula:

$$\lambda_{k+1} = (\nabla h_{k+1}^T \nabla h_{k+1})^{-1} (h_{k+1} - \nabla h_{k+1}^T \nabla f_{k+1}) . \tag{2.2.9}$$

iii) Tapia's update formula:

$$\lambda_{k+1} = (\nabla h_k^T \nabla_x^2 l_k^{-1} \nabla h_k)^{-1} [h_k - \nabla h_k^T \nabla_x^2 l_k^{-1} \nabla f_k] . \tag{2.2.10}$$

The last formula is equivalent to

$$\lambda_{k+1} = -(\nabla h_k^T \nabla h_k)^{-1} \nabla h_k^T (\nabla f_k + \nabla_x^2 l_k \, s_k) .$$

where s_k is the SQP step.

Fletcher in (1973) proposed the differentiable exact penalty function

$$\Phi(x;r) = f(x) + \lambda(x)^T h(x) + r \; || h(x) ||_2^2 ,$$

where $\lambda(x) = -(\nabla h(x)^T \nabla h(x))^{-1} \nabla h(x)^T \nabla f(x)$. It is also used as a merit function, with (2.2.8) to estimate the value of the multiplier, by Powell and Yuan (1986). This function has the advantage that when the second order sufficiency conditions are assumed and r is sufficiently large, then the minimizer of Fletcher's exact penalty function is a solution to problem (EQ). However, this function is expensive to evaluate.

Another interesting merit function is proposed by Di Pillo and Grippo (1979).

It has the form:

$$\Phi(x,\lambda;r) = f(x) + \lambda^T h(x) + r \,||\, h \,||_2^2 + \,||\, M(x)(\nabla f + \nabla h\,\lambda) \,||_2^2 \quad (2.2.11)$$

where $M(x)$ is a full rank matrix of order $m \times n$ or $n \times n$.

This function does not belong to the class of functions of the form (2.2.6). However, it is appropriate to mention it here.

If $M(x)\nabla h(x)$ is an $m \times m$ nonsingular matrix for all x, then, under some regularity and continuity assumptions, it can be shown that for sufficiently large r, all local minimizers of (2.2.11) are solutions of the problem (EQ). [See Bertsekas (1982)]

If we choose

$$M(x) = (\nabla h(x)^T \nabla h(x))^{-1} \nabla h(x)^T$$

then $M(x)\,\nabla h(x) = I$. Thus the local minimizer of (2.2.11) and Fletcher's exact penalty function are identical if r is replaced by $r - \frac{1}{4}$. So, we can regard Di Pillo and Grippo's merit function as a generalization of Fletcher's exact penalty function.

Boggs and Tolle (1984) use the following exact penalty function:

$$\Phi(x) = f(x) + \lambda(x)^T h(x) + r \,||\, A^{1/2} h(x) \,||_2^2$$

where $A(x) = (\nabla h(x)^T \nabla h(x))^{-1}$ and $\lambda(x) = -(\nabla h(x)^T \nabla h(x))^{-1} \nabla h(x)^T \nabla f(x)$.

It is quite interesting to notice that Boggs and Tolle's exact penalty function is equivalent to the Lagrangian function (1.1.1) when $\lambda(x)$ is given by the following relaxed Miele's update:

$$\lambda_r(x) = (\nabla h(x)^T \nabla h(x))^{-1} [\, r\, h(x) - \nabla h(x)^T \nabla f(x) \,] .$$

In that sense we can say that $l(x,\lambda_r(x))$ is an exact penalty function.

Boggs and Tolle's function, Fletcher's exact penalty function, and Di Pillo and Grippo's function share the disadvantage that they contain first derivatives, so their second derivatives will be either impossible or very expensive to evaluate.

Schittkowski (1983), Gill, Murray, Sunders, and Wright (1986), use as a merit function, the augmented Lagrangian (2.2.7) in which the Lagrange multiplier is treated as a separate variable.

Schittkowski (1983), Gill, Murray, Saunders, and Wright (1986) use the following scheme to update the Lagrange multiplier:

$$\lambda_+ = \alpha\,\mu + (\,1-\alpha\,)\,\lambda_c \qquad\qquad \alpha \in (0,1)$$

where $\mu = \lambda^{QP}$ and starting with $\lambda_1 = \mu_1$.

Celis, Dennis and Tapia (1984) used the augmented Lagrangian as a merit function. They fix the multiplier during the process of testing the step and update it after accepting the step.

Celis, Dennis, and Tapia (1987) used the augmented Lagrangian as a primary merit function with the function (2.2.2) as an auxiliary merit function.

In this research we will use the augmented Lagrangian as a merit function in which the Lagrange multiplier is treated as a separate variable. We will use the following formula to update the estimate of the Lagrange multiplier

$$\lambda_{k+1} = -\,(\,\nabla h_k^T \nabla h_k\,)^{-1} \nabla h_k^T (\,\nabla f_k + B_k\,\hat{s}_k\,)\,,$$

where $\hat{s}_k$ in the formula is the trial step.

2.2.2) Some Existing Methods

Problem (EQ) is often solved by the Successive Quadratic Programming

(SQP) algorithm (see Section (1.2.3)). Namely, at the k^{th} iteration the step is computed by solving the following quadratic programming subproblem:

$$\begin{aligned} \text{minimize} \quad & \nabla_x l(x_k,\lambda_k)^T s + \frac{1}{2} s^T B_k s \\ \text{subject to} \quad & h(x_k) + \nabla h(x_k)^T s = 0 , \end{aligned} \tag{QP}$$

where B_k is the Hessian of the Lagrangian or an approximation to it.

The local convergence analysis for the SQP algorithm has been fairly well established. The area of global convergence is currently receiving much attention.

Many publications have considered globally convergent algorithms, via merit functions and line searchs. [for example see Han (1977b), Fletcher (1981), Bartholomew-Biggs (1982), Schittkowski (1983), Powell and Yuan (1984), Burke and Han (1985), Boggs and Tolle (1986), Gilbert (1986) and Gill, Murray, Saunders and Wright (1986)].

Schittkowski (1983) and Gill, Murray, Saunders, and Wright (1986) solve the QP subproblem to get s^{QP} and λ^{QP}. A steplength parameter α_k is obtained by using a line search globalization strategy with the augmented Lagrangian as a merit function, then the new iterate is defined to be

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k s^{QP} \\ \lambda_{k+1} &= \lambda_k + \alpha_k (\lambda^{QP} - \lambda_k) . \end{aligned}$$

The idea behind this approach is that, since the variable x is controlled by the line search globalization strategy, the variable λ has to be controlled by a form of line search. This idea explains why they use for computing λ_{k+1} a convex combination of λ_k and λ^{QP}.

Trust region approaches for unconstrained optimization have proven to be very successful both theoretically and practically. The most natural way to intro-

duce the trust region idea to constrained optimization is to add a constraint which restricts the size of the step in problem (QP). That is, at the k^{th} iteration we solve the following trust region quadratic programming subproblem:

$$\begin{aligned} \text{minimize} \quad & \nabla_x l(x_k,\lambda_k)^T s + \frac{1}{2} s^T B_k s \\ \text{subject to} \quad & h(x_k) + \nabla h(x_k)^T s = 0 \\ & || s ||_2 \leq \Delta_k \end{aligned} \tag{TRQP}$$

However, this approach may lead to inconsistent constraints because the hyperplane $h(x_k) + \nabla h(x_k)^T s = 0$ may not intersect the sphere $|| s ||_2 \leq \Delta_k$. Even if they intersect, there is no guarantee that the trial step s will sufficiently decrease Φ and be accepted. So we may need to decrease the radius of the trust region, and again we may get inconsistent constraints if Δ_k becomes too small. Consequently, there will be no feasible region that satisfies both constraints, and the model subproblem will not have a solution in the trust region.

It is easy to overcome this difficulty if the constraints are linear (*i.e.* for general linear equality and inequality constrained optimization problem). To do this simply maintain feasibility at each iteration by either projecting or restoring the step to the feasible region. This can be done efficiently for linearly constrained optimization problems. If we do that at the k^{th} iteration the step will be computed by solving the following subproblem:

$$\begin{aligned} \text{minimize} \quad & \nabla_x l(x_k,\lambda_k)^T s + \frac{1}{2} s^T B_k s \\ \text{subject to} \quad & \nabla h(x_k)^T s = 0 \\ & || s ||_2 \leq \Delta_k , \end{aligned}$$

which has always consistent constraints. [See Gay (1983)]

For nonlinear constraints, to overcome this difficulty, two main approaches have been introduced. The first approach is to relax the constraints by

considering the following subproblem:

$$\begin{aligned} \text{minimize} \quad & \nabla_x l(x_k,\lambda_k)^T s + \frac{1}{2} s^T B_k s \\ \text{subject to} \quad & \alpha\, h(x_k) + \nabla h(x_k)^T s = 0 \\ & || s ||_2 \leq \Delta_k \end{aligned}$$

where $0 \leq \alpha \leq 1$. This approach has been applied by Vardi (1985) and Byrd, Schnabel, and Shultz (1985).

Using this approach makes the problem always feasible in the sense that if we set $\alpha = 0$ then the hyperplane $\alpha\, h(x_k) + \nabla h(x_k)^T s = 0$ will contain the current point and consequently it will intersect with a trust region sphere of any radius. However, this approach suffers from a disadvantage that the step depends on the unknown parameter α which there is no clear way of choosing.

An interesting way using this approach to compute a trial step that does not depend on the parameter α was implemented by Byrd, Omojokun, Schnabel, and Shultz (1987). They calculate s by solving the following subproblem

$$\begin{aligned} \text{minimize} \quad & \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s \\ \text{subject to} \quad & \nabla h(x_k)^T s = \nabla h(x_k)^T v \\ & || s ||_2 \leq \Delta_k , \end{aligned}$$

where v solves the following problem

$$\begin{aligned} \text{minimize} \quad & || h(x_k) + \nabla h(x_k)^T v ||_2 \\ \text{subject to} \quad & || v ||_2 \leq \xi \Delta_k , \end{aligned}$$

where $0 < \xi < 1$.

The second approach is to add the trust region constraint to a somewhat different problem. At the k^{th} iteration the step is taken to be the one which minimizes the quadratic model of the Lagrangian and gives some decrease in

$|| h_k + \nabla h_k^T s ||_2$. This idea was first introduced by Celis, Dennis, and Tapia (1984). At each iteration the step is computed by solving the following subproblem:

$$\begin{aligned} \text{minimize} \quad & \nabla_x l(x_k,\lambda_k)^T s + \frac{1}{2} s^T B_k s \\ \text{subject to} \quad & || h(x_k) + \nabla h(x_k)^T s ||_2 \leq \theta_k \\ & || s ||_2 \leq \Delta_k \end{aligned} \qquad \text{(CDT)}$$

where θ_k is some positive constant that depends on k.

Celis, Dennis and Tapia (1984) chose θ_k to be $|| h_k + \nabla h_k^T s_k^{cp} ||_2$, where $s_k^{cp} = -\alpha_k \nabla h_k h_k$ is the step to the Cauchy point, *i.e.*, the minimizer in the trust region $\{s : || s ||_2 \leq \Delta_k\}$ of $|| h(x_k) + \nabla h(x_k)^T s ||_2$ along its negative gradient. That is, the Celis-Dennis-Tapia step is chosen from the set of steps from x_k that are inside the trust region and give at least as much descent on the 2-norm of the residual of the linearized constraints as the Cauchy step.

In 1986 Powell and Yuan introduced a different way of choosing θ_k. They chose it to be any number that satisfies

$$\theta_k = \min [|| h(x_k) + \nabla h(x_k)^T s ||_2 : || s ||_2 \leq \sigma\Delta_k, 0 \leq \sigma \leq 1].$$

For any choice of θ_k, if s solves the CDT subproblem, then

$$(B_k + \mu I + \alpha \nabla h_k \nabla h_k^T) s = -(\nabla_x l_k + \alpha \nabla h_k h_k), \qquad (2.2.12)$$

$$|| s ||_2 \leq \Delta_k ,$$

$$\mu (\Delta_k - || s ||_2) = 0 ,$$

$$|| h(x_k) + \nabla h(x_k)^T s ||_2 \leq \theta_k ,$$

$$\alpha (\theta_k - || h(x_k) + \nabla h(x_k)^T s ||_2) = 0 ,$$

with μ , $\alpha \geq 0$.

The approach of Fletcher (1984) is different. This approach uses an l_1 exact penalty function with a trust region constraint. Let the linearized constraints be $\hat{l}(s)$ and the quadratic model of the Lagrangian be $q(s)$, then the l_1 exact penalty function is formed as follows

$$l_1(s,\mu) = q(s) + \sum_{i=1}^{m} \mu_i \hat{l}_i(s)$$

At each iteration the step is computed by minimizing this l_1 exact penalty function subject to a trust region constraint.

CHAPTER THREE

THE TRUST REGION ALGORITHM

This chapter is devoted to presenting in detail a variant of the 1984 Celis-Dennis-Tapia trust region algorithm for equality constrained optimization problem. Before we start our discussion about the algorithm, let us introduce some of the notation that will be used in the rest of this thesis.

Notation

The trial step at the k^{th} iteration is denoted by $\hat{s}_k$ and its associated Lagrange multiplier by $\hat{\Delta\lambda}_k$. If the step is accepted it will be denoted by s_k and its associated Lagrange multiplier by $\Delta\lambda_k$.

The terms $\nabla^2 h(x_k)\,\Delta\lambda$ and $\nabla^2 h(x_k)\,h(x_k)$ will appear in Chapter 4 and 5. They are used to denote $\sum_{i=1}^{m} \nabla^2 h_i(x_k)\,\Delta\lambda_i$ and $\sum_{i=1}^{m} \nabla^2 h_i(x_k)\,h_i(x_k)$ respectively.

The matrix B_k denotes $\nabla_x^2 l(x_k,\lambda_k)$ or an approximation to it.

3.1 DESCRIPTION OF THE ALGORITHM

The algorithm is iterative. At each iteration a trial step $\hat{s}_k$ is obtained by solving a model problem.

At any iteration indexed k , we try to update the estimate of the solution

x_k to be the improved estimate x_{k+1}. To do this, the step s_k^{QP} is computed by solving the QP subproblem. If it exists and lies inside the trust region, *i.e.* if $|| s_k^{QP} || \leq \Delta_k$, then we set $\hat{s}_k = s_k^{QP}$. Otherwise, the CDT subproblem will be solved. On the other hand, if x_k is feasible, then we solve the TRQP subproblem. This can be stated as an algorithm as follows

ALGORITHM (3.1.1) Computing the Trial Step

Solve (QP) to get s_k^{QP} and $\Delta\lambda_k^{QP}$

If $|| s_k^{QP} ||_2 \leq \Delta_k$

then $\hat{s}_k = s_k^{QP}$

$$\hat{\Delta\lambda}_k = \Delta\lambda_k^{QP}.$$

Else, if x_k is feasible

then solve (TRQP)

Set $\hat{s}_k = s_k^{TRQP}$

$$\hat{\Delta\lambda}_k = -(\nabla h_k^T \nabla h_k)^{-1} \nabla h_k^T [\nabla_x l_k + B_k s_k^{TRQP}].$$

Else, solve (CDT)

Set $\hat{s}_k = s_k^{CDT}$.

$$\hat{\Delta\lambda}_k = -(\nabla h_k^T \nabla h_k)^{-1} \nabla h_k^T [\nabla_x l_k + B_k s_k^{CDT}].$$

Let $\hat{s}_k$ be the step computed by the algorithm and $\hat{\Delta\lambda}_k$ be the corresponding Lagrange multiplier step, we test whether the point $(x_k + \hat{s}_k , \lambda_k + \hat{\Delta\lambda}_k)$ is a better approximation to the solution (x_* , λ_*). In order to do this, we use, as a

merit function, the augmented Lagrangian (2.2.7).

Now, we test $(x_k+\hat{s}_k , \lambda_k+\hat{\Delta\lambda}_k)$ to determine whether it makes an improvement in the merit function.

We define the actual reduction in the merit function in going from (x_k,λ_k) to $(x_k+\hat{s}_k , \lambda_k+\hat{\Delta\lambda}_k)$ by:

$$\begin{aligned} Ared_k &= L(x_k,\lambda_k;r_k) - L(x_k+\hat{s}_k,\lambda_k+\hat{\Delta\lambda}_k;r_k) \\ &= l(x_k,\lambda_k) - l(x_k+\hat{s}_k,\lambda_k+\hat{\Delta\lambda}_k) + r_k \left[\| h(x_k) \|_2^2 - \| h(x_k+\hat{s}_k) \|_2^2 \right]. \end{aligned}$$

Which also can be written as:

$$\begin{aligned} Ared_k &= l(x_k,\lambda_k) - l(x_k+\hat{s}_k,\lambda_k) - \hat{\Delta\lambda}_k^T h(x_k+\hat{s}_k) \\ &+ r_k \left[\| h(x_k) \|_2^2 - \| h(x_k+\hat{s}_k) \|_2^2 \right]. \end{aligned} \tag{3.1.1}$$

The step $\hat{s}_k$ calculation is based on a quadratic approximation of the Lagrangian function and a linear approximation to the constraints. Now by using the same approximation we can compute the predicted reduction which is defined by

$$Pred_k = L(x_k,\lambda_k;r_k) - \Psi(x_k,\hat{s}_k,\lambda_k,\hat{\Delta\lambda}_k;r_k),$$

where $\Psi(x_k,\hat{s}_k,\lambda_k,\hat{\Delta\lambda}_k;r_k)$ is an approximation to $L(x_k+\hat{s}_k,\lambda_k+\hat{\Delta\lambda}_k;r_k)$ and is defined by:

$$\begin{aligned} \Psi(x_k,\hat{s}_k,\lambda_k,\hat{\Delta\lambda}_k;r_k) &= l(x_k,\lambda_k) + \nabla_x l(x_k,\lambda_k)^T \hat{s}_k + \frac{1}{2}\hat{s}_k{}^T B_k \hat{s}_k \\ &+ \hat{\Delta\lambda}_k^T [h(x_k)+\nabla h(x_k)^T \hat{s}_k] \\ &+ r_k \| h(x_k) + \nabla h(x_k)^T \hat{s}_k \|_2^2 . \end{aligned}$$

Hence,

$$\begin{aligned} Pred_k &= L(x_k,\lambda_k;r_k) - [l(x_k,\lambda_k) + \nabla l (x_k,\lambda_k)^T \hat{s}_k + \frac{1}{2} \hat{s}_k^T B_k \hat{s}_k] \\ &- \hat{\Delta\lambda}_k^T (h(x_k) + \nabla h(x_k)^T \hat{s}_k) \end{aligned}$$

$$- \; r_k \; || \; h(x_k) + \nabla h(x_k)^T \hat{s}_k \; ||_2^2 \; ;$$

which can be written as:

$$Pred_k \; = \; - \nabla_x l(x_k,\lambda_k)^T \hat{s}_k - \frac{1}{2} \; \hat{s}_k{}^T B_k \; \hat{s}_k - \hat{\Delta\lambda}_k^T(\; h(x_k)+\nabla h(x_k)^T \hat{s}_k \;)$$
$$+ \; r_k \; [\; || \; h(x_k) \; ||_2^2 - \; || \; h(x_k) + \nabla h(x_k)^T \hat{s}_k \; ||_2^2 \;] \; . \qquad (3.1.2)$$

We accept the step and set $x_{k+1} = x_k + s_k$ and $\lambda_{k+1} = \lambda_k + \Delta\lambda_k$, if

$$\frac{Ared_k}{Pred_k} \; \geq \; \eta_1$$

where $\eta_1 \; \epsilon \; (0,1)$ is a small fixed constant.

If the step is rejected, then we set $x_{k+1} = x_k$ and $\lambda_{k+1} = \lambda_k$ and decrease the radius of the trust region by setting

$$\Delta_{k+1} \; \epsilon \; [\; \alpha_1 \; || \; \hat{s}_k \; ||_2 \; , \; \alpha_2 \; || \; \hat{s}_k \; ||_2 \;] \; ,$$

where $0 < \alpha_1 \leq \alpha_2 < 1$. [See Dennis and Schnabel (1983)].

When the step is accepted, 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 $\eta_2 \; \epsilon \; (\eta_1,1)$, then the radius of the trust region is updated by the rule:

$$\Delta_{k+1} \; = \; \min \; [\; \Delta_k \; , \; \alpha_3 \; || \; s_k \; ||_2 \;]$$

where $\alpha_3 > 1$.

However, if $\frac{Ared_k}{Pred_k} \; \geq \; \eta_2$, then we increase the radius of the trust region by setting:

$$\Delta_{k+1} = \max [\ \Delta_k\ ,\ \alpha_3\ ||\ s_k\ ||_2\ ]\ .$$

This can be stated as an algorithm as follows:

ALGORITHM (3.1.2) Testing the Step and Updating the Trust Region Radius

If $\dfrac{Ared_k}{Pred_k} < \eta_1$,

then set $x_{k+1} = x_k$,

$\lambda_{k+1} = \lambda_k$,

$\Delta_{k+1} \in [\ \alpha_1\ ||\ \hat{s}_k\ ||_2\ ,\ \alpha_2\ ||\ \hat{s}_k\ ||_2\ ]$. (3.1.3)

Else, if $\eta_1 \leq \dfrac{Ared_k}{Pred_k} < \eta_2$

then set $x_{k+1} = x_k + s_k$,

$\lambda_{k+1} = \lambda_k + \Delta\lambda_k$,

$\Delta_{k+1} = \min\ [\Delta_k\ ,\ \alpha_3\ ||\ s_k\ ||_2\ ]$.

Else, if $\dfrac{Ared_k}{Pred_k} \geq \eta_2$ and $||\ s^{QP}\ ||_2 > \Delta_k$ and $\alpha_4\ ||\ s_k\ ||_2 > \Delta_k$

then we do only one internal doubling according to algorithm (3.1.3) below.

Else, set $x_{k+1} = x_k + s_k$,

$\lambda_{k+1} = \lambda_k + \Delta\lambda_k$,

$\Delta_{k+1} = \max[\ \Delta_k\ ,\ \alpha_3\ ||\ s_k\ ||_2\ ]$.

In the case when $\frac{Ared_k}{Pred_k} \geq \eta_2$, $s_k \neq s^{QP}$, and $\alpha_4 \mid\mid s_k \mid\mid_2 > \Delta_k$, where $\alpha_4 > 1$, then we do only one internal doubling by setting $\Delta_k := \alpha_4 \mid\mid s_k \mid\mid_2$ and if $\mid\mid s^{QP} \mid\mid_2 < \Delta_k$, we take it as our trial step. Otherwise, we stay with the old acceptable step and update the old trust region radius by the rule

$$\Delta_{k+1} = \max [\, \Delta_k \, , \alpha_3 \mid\mid s_k \mid\mid_2 \,] \, .$$

This can be stated as an algorithm as follows

ALGORITHM (3.1.3) Internal Doubling

Set $\Delta_k = \alpha_4 \mid\mid s_k \mid\mid_2$.

If $\mid\mid s_k^{QP} \mid\mid_2 > \Delta_k$,

then, go back to the last acceptable step and the last corresponding trust region radius and update it by step (4) of algorithm (3.1.2).

Else, if $\frac{Ared_k}{Pred_k} < \eta_1$,

then go back to the last acceptable step and the last corresponding trust region radius and update it by step (4) of algorithm (3.1.2).

Else, accept the step and update Δ_k according to step (1) or (4) of algorithm (3.1.2) above.

3.2 THE ALGORITHM

The outline of the algorithm is given below. It differs from the 1984 Celis-Dennis-Tapia algorithm in its way of updating the penalty parameter in step (3)

of the algorithm and in its way of updating of the trust region radius in step (4).

Step (0)

Set $x_0 \in R^n$, $B_0 \in R^{n\times n}$, $\lambda_0 \in R^m$,

$r_{-1} = 1$, $\rho > 0$,

$0 < \alpha_1 \leq \alpha_2 < 1 < \alpha_4 \leq \alpha_3$,

$0 < \eta_1 < \eta_2 < 1$,

$\epsilon > 0$, $\Delta_0 > 0$,

and $k = 0$.

Step (1)

If $|| P_k \nabla f_k ||_2 + || h_k ||_2 < \epsilon$, stop.

Step (2)

Compute $\hat{s}_k$ and $\Delta\hat{\lambda}_k$ according to algorithm (3.1.1) above.

Step (3)

Update the penalty parameter by the following scheme:

Set $r_k = r_{k-1}$

If

$$Pred_k \geq \frac{r_k}{2}[\, || h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2 \,]$$

go to step (4)

Else, set

$$r_k = 2 \frac{\nabla_x l_k^T \hat{s}_k + \frac{1}{2} \hat{s}_k^T B_k \hat{s}_k + \hat{\Delta\lambda}_k^T (h_k + \nabla h_k^T \hat{s}_k)}{|| h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2} + \rho .$$

Step (4)

Test the step and update Δ_k according to algorithm (3.1.2) above.

Step (5)

Set $k := k + 1$ and go to step (1).

CHAPTER FOUR

GLOBAL CONVERGENCE ANALYSIS

This chapter is devoted to the analysis of the global behavior of our algorithm. Our global convergence theory is sufficiently general that it holds for any algorithm that generates steps that give at least a fraction of Cauchy decrease in the quadratic model of the constraints.

In the first part of this chapter we state the standard assumptions under which the global convergence theory is proven. In the rest of the chapter we address the global convergence theory of the algorithm. In Section 4.2 we prove lemmas that deal with the predicted decrease of the function and of the model. In Section 4.3 we prove lemmas that are needed to study the behavior of the penalty parameter. Section 4.4 is devoted to studying the global convergence analysis of the algorithm.

4.1 THE STANDARD ASSUMPTIONS

It is clear that the behavior of our algorithm will depend on the conditions we impose on the problem and on the matrices. We first state our assumptions

1) There exists an open convex set $\Omega \in R^n$ such that, for all k, x_k and $x_k + \hat{s}_k \in \Omega$.

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

3) There exists a positive constant Δ_* such that, for all k, $\Delta_k \leq \Delta_*$.

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

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

6) The matrices $\{ B_k , k=1,2,...\}$ have a uniform upper bound.

Remark

Assumption (3) implies that all the trial steps are bounded. This assumption is not a restrictive assumption. In fact, in our convergence theory we never state that the radius of the trust region has to be increased. So we can set an upper bound on the radius of the trust region inside the algorithm and our global convergence theory holds.

4.2 SUFFICIENT DECREASE IN THE MODEL

All results in this section deal with the reduction of the merit function and the predicted reduction of the model.

In the following lemma we use the fact that the step $\hat{s}_k$ is chosen to give at least as much decrease in the linearization of the constraints as the Cauchy step s_k^{cp}.

Lemma (4.1)

Let $\hat{s}_k$ be the step generated by the algorithm. Then there exist constants b_1

and b_2 such that for all k

$$|| h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2 \geq \frac{|| h_k ||_2}{b_1} \min [\Delta_k , \frac{|| h_k ||_2}{b_2}] .$$

Proof

From the way of computing the step $\hat{s}_k$, we have

$$\begin{aligned} || h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2 &\geq || h_k ||_2^2 - \theta_k^2 \\ &= || h_k ||_2^2 - || h_k + \nabla h_k^T s_k^{cp} ||_2^2 \\ &= -2 h_k^T \nabla h_k^T s_k^{cp} - (s_k^{cp})^T \nabla h_k \nabla h_k^T s_k^{cp} . \end{aligned}$$

From the definition of s_k^{cp} , we have

$$s_k^{cp} = - \alpha_k \nabla h_k h_k ,$$

where α_k is defined by

$$\alpha_k = \frac{\Delta_k}{|| \nabla h_k h_k ||_2} \quad \text{if} \quad \frac{|| \nabla h_k h_k ||_2^3}{|| \nabla h_k^T \nabla h_k h_k ||_2^2} \geq \Delta_k , \qquad (4.2.1\text{-a})$$

otherwise,

$$\alpha_k = \frac{|| \nabla h_k h_k ||_2^2}{|| \nabla h_k^T \nabla h_k h_k ||_2^2} . \qquad (4.2.1\text{-b})$$

Consider the first case. *i.e.*, the case when $s_k^{cp} = - \Delta_k \frac{\nabla h_k h_k}{|| \nabla h_k h_k ||_2}$. In this case, using $\frac{|| \nabla h_k h_k ||_2^3}{|| \nabla h_k^T \nabla h_k h_k ||_2^2} \geq \Delta_k$, we have

$$\begin{aligned} || h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2 &\geq 2 \Delta_k || \nabla h_k h_k ||_2 - \Delta_k^2 \frac{|| \nabla h_k^T \nabla h_k h_k ||_2^2}{|| \nabla h_k h_k ||_2^2} \\ &\geq 2 \Delta_k || \nabla h_k h_k ||_2 - \Delta_k || \nabla h_k h_k ||_2 \end{aligned}$$

$$= \Delta_k \; || \nabla h_k \; h_k \; ||_2 \, . \qquad (4.2.2)$$

Now, consider the second case. We have

$$|| h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2 \geq 2 \frac{|| \nabla h_k \; h_k ||_2^2}{|| \nabla h_k^T \nabla h_k \; h_k ||_2^2} \; || \nabla h_k \; h_k ||_2^2$$
$$- \left[\frac{|| \nabla h_k \; h_k ||_2^2}{|| \nabla h_k^T \nabla h_k \; h_k ||_2^2} \right]^2 \; || \nabla h_k^T \nabla h_k \; h_k ||_2^2 \, .$$

Hence,

$$|| h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2 \geq \frac{|| \nabla h_k \; h_k ||_2^4}{|| \nabla h_k^T \nabla h_k \; h_k ||_2^2}$$

$$\geq \frac{|| \nabla h_k \; h_k ||_2^2}{|| \nabla h_k \; \nabla h_k^T ||_2} \, . \qquad (4.2.3)$$

From (4.2.2) and (4.2.3), we can write

$$|| h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2 \geq || \nabla h_k \; h_k ||_2 \min \left[\Delta_k \, , \frac{|| \nabla h_k \; h_k ||_2}{|| \nabla h_k \nabla h_k^T ||_2} \right] .$$

Now, using the standard assumptions, since

$$|| \nabla h_k \; h_k ||_2 \geq \frac{|| h_k ||_2}{|| (\nabla h_k^T \nabla h_k)^{-1} \nabla h_k^T ||_2} \, ,$$

we can write

$$|| h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2 \geq$$

$$\frac{|| h_k ||_2}{|| (\nabla h_k^T \nabla h_k)^{-1} \nabla h_k^T ||_2} \min \left[\Delta_k \, , \frac{|| h_k ||_2}{|| (\nabla h_k^T \nabla h_k)^{-1} \nabla h_k^T ||_2 \, || \nabla h_k \nabla h_k^T ||_2} \right] \qquad (4.2.4)$$

Now from the standard assumptions there exist constants b_1, and b_2 where

$$b_1 = \sup_{x \, \epsilon \, \Omega} || (\nabla h(x)^T \nabla h(x))^{-1} \nabla h(x)^T ||_2$$

and

$$b_2 = \sup_{x \in \Omega} [\ || (\nabla h(x)^T \nabla h(x))^{-1} \nabla h(x)^T ||_2 \ || \nabla h(x) \nabla h(x)^T ||_2 \ ]$$

The rest of the proof follows immediately by substituting b_1 and b_2 into (4.2.4). ■

Corollary (4.2)

Let k be the index of any iteration, then the predicted decrease in the model by the trial step satisfies

$$Pred_k \geq \frac{r_k}{2} \frac{|| h_k ||_2}{b_1} \min [\ \Delta_k\ , \frac{|| h_k ||_2}{b_2}\ ],$$

where b_1 and b_2 are as in Lemma (4.1).

Proof

From the way of updating the penalty parameter r_k in step 3 of the algorithm, we have

$$Pred_k \geq \frac{r_k}{2} [\ || h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2\ ].$$

The rest of the proof follows immediately from the last lemma. ■

Lemma (4.1) shows that the way of choosing θ_k in the CDT subproblem implies that we always get a fraction of Cauchy decrease in the constraints.

Corollary (4.2) shows that the way we update the penalty parameter insures that the predicted reduction at each iteration will be at least as much as a fraction of Cauchy decrease.

Lemma (4.3)

If s is the solution to the following problem

$$\begin{aligned} &\text{minimize} \quad g^T s + \frac{1}{2} s^T B\, s \\ &\text{subject to} \quad \| s \|_2 \leq \Delta \end{aligned}$$

for any $g \in R^n$ and any $n \times n$ symmetric matrix B, then

$$g^T s \leq -\frac{1}{2} \| g \|_2 \min [\, \Delta, \frac{\| g \|_2}{2 \| B \|_2} \,]. \tag{4.2.5}$$

Proof

The proof follows directly from Lemma (3.2) of Powell and Yuan (1986). However, for the sake of completeness we present a proof for the lemma.

If $\| g \|_2 = 0$, then (4.2.5) is trivial. So, let us consider the case when $\| g \|_2 > 0$.

If the trust region is not active then the step is computed from $B\, s = -g$. Hence we can write

$$s = B^+ g + \bar{s},$$

where B^+ is the generalized inverse of B and $\bar{s}$ is a vector in the null space of B.

Since g is in the range space of B, it follows that

$$\begin{aligned} g^T s &= -g^T B^+ g \\ &\leq -\frac{1}{\| B \|_2} \| g \|_2^2 . \\ &\leq -\frac{1}{4 \| B \|_2} \| g \|_2^2 . \end{aligned} \tag{4.2.6}$$

If the trust region is active then from Kuhn-Tucker theory there exists a multiplier $\mu \geq 0$ such that

$$g + (B + \mu I) s = 0 \tag{4.2.7}$$

Using the same argument as above, we can write

$$g^T s \leq - \frac{1}{|| B + \mu I ||_2} || g ||_2^2 .$$

But from (4.2.7), we have

$$\mu || s ||_2 = || B s + g ||_2 \leq || B ||_2 || s ||_2 + || g ||_2 ,$$

or

$$\mu \leq || B ||_2 + \frac{|| g ||_2}{\Delta} .$$

Thus, we have

$$\begin{aligned} || B + \mu I ||_2 &\leq || B ||_2 + \mu \\ &\leq 2 || B ||_2 + \frac{|| g ||_2}{\Delta} \\ &= \frac{2 || B ||_2 \Delta + || g ||_2}{\Delta} . \end{aligned}$$

So,

$$\begin{aligned} g^T s &\leq - \frac{|| g ||_2^2}{|| B + \mu I ||_2} \\ &\leq - \frac{|| g ||_2^2 \Delta}{2 || B ||_2 \Delta + || g ||_2} \end{aligned}$$

From the last inequality and (4.2.6), we can write

$$\leq - \frac{1}{2} || g ||_2 \min [\Delta , \frac{|| g ||_2}{2 || B ||_2}] .$$

Hence we get the desired result. ■

Corollary (4.4)

For any step $\hat{s}_k$ generated by the algorithm, let $\hat{s}_k^p = P_k\ \hat{s}_k$ and $\hat{s}_k^q = Q_k\ \hat{s}_k$ where $P_k = I - \nabla h_k(\nabla h_k^T \nabla h_k)^{-1}\nabla h_k^T$ and $Q_k = I - P_k$. Then, $\hat{s}_k^p$ solves the following problem:

$$\begin{aligned} \text{minimize} \quad & [P_k\ (\nabla l_k + B_k\ \hat{s}_k^q\ )]^T s + \frac{1}{2}\ s^T P_k B_k P_k\ s \\ \text{subject to} \quad & \|\ s\ \|_2 \leq \overline{\Delta}_k \end{aligned}$$

where $\overline{\Delta}_k = \sqrt{\Delta_k^2 - \|\ \hat{s}_k^q\ \|^2}$. Furthermore, $\hat{s}_k^p$ satisfies:

$$(\ \nabla l_k + B_k\ \hat{s}_k^q\ )^T \hat{s}_k^p \leq -\frac{1}{2}\ \|\ P_k(\nabla l_k + B_k\ \hat{s}_k^q\ )\ \|_2 \min\ [\ \overline{\Delta}_k\ ,\ \frac{\|\ P_k(\nabla l_k + B_k\ \hat{s}_k^q\ )\ \|_2}{2\ \|\ B_k\ \|_2}\ ].$$

Proof

The proof follows directly from Powell and Yuan (1986). However, for the sake of completeness we present a proof for this lemma.

Since $\hat{s}_k = \hat{s}_k^p + \hat{s}_k^q$, $\hat{s}_k^p$ solves the following problem:

$$\begin{aligned} \text{minimize} \quad & \nabla l_k^T(s + \hat{s}_k^q\ ) + \frac{1}{2}\ (s + \hat{s}_k^q\ )^T B_k\ (s + \hat{s}_k^q\ ) \\ \text{subject to} \quad & \nabla h_k^T s = 0 \\ & \|\ s + \hat{s}_k^q\ \|_2 \leq \Delta_k\ . \end{aligned}$$

The last problem is equivalent to

$$\begin{aligned} \text{minimize} \quad & (\nabla l_k + B_k\ \hat{s}_k^q\ )^T s + \frac{1}{2}\ s^T B_k\ s \\ \text{subject to} \quad & \nabla h_k^T s = 0 \\ & \|\ s\ \|_2 \leq \overline{\Delta}_k\ . \end{aligned}$$

Since, $\hat{s}_k^p$ lies in the null space of ∇h_k^T , then $\nabla h_k^T \hat{s}_k^p$ is always zero. Hence $\hat{s}_k^p$

solves the last problem even if the constraint $\nabla h_k^T s = 0$ is deleted. That is, $\hat{s}_k^p$ solves the following problem.

$$\begin{aligned} \text{minimize} \quad & [P_k(\nabla l_k + B_k \hat{s}_k^q)]^T s + \frac{1}{2} s^T P_k B_k P_k s \\ \text{subject to} \quad & \| s \|_2 \leq \overline{\Delta}_k . \end{aligned}$$

Now using (4.2.5) and $\| P_k B_k P_k \|_2 \leq \| B_k \|_2$, we get

$$(\nabla l_k + B_k \hat{s}_k^q)^T \hat{s}_k^p \leq -\frac{1}{2} \| P_k(\nabla l_k + B_k \hat{s}_k^q) \|_2 \min [\overline{\Delta}_k , \frac{\| P_k(\nabla l_k + B \hat{s}_k^q) \|_2}{2 \| B_k \|_2}].$$

Hence we get the desired result. ■

Lemma (4.5)

There exists a constant c_1 such that $\| \nabla l_k \|_2 \leq c_1$.

Proof

Since $\lambda_k = \lambda_{k-t_k+1} = -(\nabla h_{k-t_k}^T \nabla h_{k-t_k})^{-1} \nabla h_{k-t_k}^T (\nabla f_{k-t_k} + B_{k-t_k} s_{k-t_k})$, where s_{k-t_k} is the last acceptable step, we have

$$\| \lambda_k \|_2 \leq \| (\nabla h_{k-t_k}^T \nabla h_{k-t_k})^{-1} \nabla h_{k-t_k}^T \|_2 [\| \nabla f_{k-t_k} \|_2 + \| B_{k-t_k} \|_2 \| s_{k-t_k} \|_2].$$

The boundedness of $\| \lambda_k \|_2$ follows immediately from the standard assumptions.

Now, because $\| \nabla l_k \|_2 \leq \| \nabla f_k \|_2 + \| \nabla h_k \|_2 \| \lambda_k \|_2$, we can see that the proof of the lemma follows from the boundedness of $\| \lambda_k \|_2$ and the standard assumptions. ■

Lemma (4.6)

For any x_k , $x_k+\hat{s}_k \in \Omega$, we have:

$$| Ared_k - Pred_k | \leq a_1 || \hat{s}_k ||_2^2 + r_k [a_2 || \hat{s}_k ||_2^3 + a_3 || h_k ||_2 || \hat{s}_k ||_2^2] ,$$

where a_1 , a_2 , a_3 are constants independent of k .

Proof

From (3.1.1) and (3.1.2) we can write:

$$\begin{aligned} Ared_k - Pred_k &= [l(x_k,\lambda_k) + \nabla_x l(x_k,\lambda_k)^T \hat{s}_k + \frac{1}{2}\hat{s}_k^T B_k \hat{s}_k - l(x_k+\hat{s}_k , \lambda_k)] \\ &+ \hat{\Delta\lambda}_k^T [h_k + \nabla h_k^T \hat{s}_k - h(x_k+\hat{s}_k)] \\ &+ r_k [|| h_k + \nabla h_k^T \hat{s}_k ||_2^2 - || h(x_k+\hat{s}_k) ||_2^2] . \end{aligned}$$

So,

$$\begin{aligned} | Ared_k - Pred_k | \leq & | l(x_k,\lambda_k) + \nabla_x l(x_k,\lambda_k)^T \hat{s}_k + \frac{1}{2} \hat{s}_k^T B_k \hat{s}_k - l(x_k+\hat{s}_k , \lambda_k) | \\ &+ | \hat{\Delta\lambda}_k^T [h_k + \nabla h_k^T \hat{s}_k - h(x_k+\hat{s}_k)] | \\ &+ r_k | || h_k + \nabla h_k^T \hat{s}_k ||_2^2 - || h(x_k+\hat{s}_k) ||_2^2 | . \end{aligned}$$

Hence,

$$\begin{aligned} | Ared_k - Pred_k | \leq & \frac{1}{2} | \hat{s}_k^T [B_k - \nabla_x^2 l(x_k+\xi_1\hat{s}_k,\lambda_k)] \hat{s}_k | \\ &+ \frac{1}{2} | \hat{s}_k^T [\nabla^2 h(x_k+\xi_2\hat{s}_k) \hat{\Delta\lambda}_k] \hat{s}_k | \\ &+ r_k | \hat{s}_k^T [\nabla h_k \nabla h_k^T - \nabla h(x_k+\xi_3\hat{s}_k) \nabla h^T(x_k+\xi_3\hat{s}_k)] \hat{s}_k | \\ &+ r_k | \hat{s}_k^T \nabla^2 h(x_k+\xi_3\hat{s}_k) h(x_k+\xi_3\hat{s}_k) \hat{s}_k | , \end{aligned}$$

for some ξ_1 , ξ_2 , $\xi_3 \in (0 , 1)$. So,

$$\begin{aligned} | Ared_k - Pred_k | \leq & \frac{1}{2} (|| \nabla_x^2 l(x_k+\xi_1\hat{s}_k,\lambda_k) ||_2 + || B_k ||_2) || \hat{s}_k ||_2^2 \\ &+ \frac{1}{2} || \nabla^2 h(x_k+\xi_2\hat{s}_k) \hat{\Delta\lambda}_k ||_2 || \hat{s}_k ||_2^2 \end{aligned}$$

$$+ r_k \; || \; \nabla h_k \; \nabla h_k^T - \nabla h(x_k+\xi_3\hat{s}_k)\nabla h^T(x_k+\xi_3\hat{s}_k) \; ||_2 \; || \; \hat{s}_k \; ||_2^2$$

$$+ r_k \; || \; \nabla^2 h(x_k+\xi_3\hat{s}_k)h(x_k+\xi_3\hat{s}_k) \; ||_2 \; || \; \hat{s}_k \; ||_2^2$$

Now by using the standard assumptions, we get

$$| \; Ared_k - Pred_k \; | \; \leq \; a_1 \; || \; \hat{s}_k \; ||_2^2 \; + \; a_2 \, r_k \; || \; \hat{s}_k \; ||_2^3 \; + \; a_3 \, r_k \; || \; \hat{s}_k \; ||_2^2 \; || \; h_k \; ||_2$$

Hence we get the desired result. ■

The result we obtained in the last lemma does not depend on any property of the matrices $\{ B_k \}$ except that they are bounded, and does not depend on any property of the step.

Corollary (4.7)

Under the assumption of Lemma (4.6), we have

$$| \; Ared_k - Pred_k \; | \; \leq \; a_0 \, r_k \; || \; \hat{s}_k \; ||_2^2$$

where a_0 is a constant independent of k .

Proof

The proof follows immediately from the last lemma, the fact that $r_k \geq 1$, and the standard assumptions. ■

Corollary (4.7) shows that our definition of predicted reduction of the merit function gives an approximation to the merit function that is accurate to within the square of the steplength.

Lemma (4.8)

If $\hat{s}_k$ and $\hat{s}_k^p$ are as in Corollary (4.4), then

$$(\nabla l_k + B_k \hat{s}_k)^T \hat{s}_k^p \leq 0 .$$

Proof

If $\hat{s}_k$ is the step generated from the CDT subproblem, then from (2.2.12) $\hat{s}_k$ satisfies

$$[B_k + \mu I + \alpha \nabla h_k \nabla h_k^T] \hat{s}_k = -\nabla l_k - \alpha \nabla h_k h_k .$$

Equivalently,

$$-(\nabla l_k + B_k \hat{s}_k) = \mu \hat{s}_k + \alpha \nabla h_k (h_k + \nabla h_k^T \hat{s}_k) .$$

Now

$$-P_k (\nabla l_k + B_k \hat{s}_k) = \mu P_k \hat{s}_k + \alpha P_k [\nabla h_k (h_k + \nabla h_k^T \hat{s}_k)] .$$

Since $P_k \nabla h_k = 0$, we get

$$-P_k (\nabla l_k + B_k \hat{s}_k) = \mu \hat{s}_k^P .$$

So, since $P_k = P_k^T$ and $P_k \hat{s}_k^p = \hat{s}_k^p$,

$$-(\nabla l_k + B_k \hat{s}_k)^T \hat{s}_k^p = \mu \| \hat{s}_k^p \|_2^2 ,$$

which implies that

$$(\nabla l_k + B_k \hat{s}_k)^T \hat{s}_k^p \leq 0 .$$

Now, assume that the step is generated from the TRQP subproblem. Then $\hat{s}_k$ must satisfy

$$(B_k + \mu I)^T \hat{s}_k = -(\nabla l_k + \nabla h_k \hat{\Delta\lambda}_k) .$$

Notice that $\mu = 0$ if the step is generated from the QP subproblem, *i.e.* if the trust region constraint is not binding. The last equation can be written as

$$\nabla l_k + B_k \hat{s}_k = - \nabla h_k \hat{\Delta\lambda}_k - \mu \hat{s}_k .$$

Now, by multiplying by P_k , we obtain

$$P_k(\nabla l_k + B_k \hat{s}_k) = -P_k \nabla h_k \hat{\Delta\lambda}_k - \mu P_k \hat{s}_k .$$

Again since $P_k \nabla h_k = 0$ we have

$$P_k (\nabla l_k + B_k \hat{s}_k) = - \mu \hat{s}_k^p ;$$

which implies that

$$(\nabla l_k + B_k \hat{s}_k)^T \hat{s}_k^p = - \mu || \hat{s}_k^p ||_2^2 \leq 0 .$$

Finally, assume that the step $\hat{s}_k = s_k^{cp} = - \alpha_k \nabla h_k h_k$, where α_k is defined by (4.2.1), then

$$\hat{s}_k^p = P_k \hat{s}_k = - \alpha_k P_k \nabla h_k h_k = 0 .$$

So,

$$(\nabla l_k + B_k \hat{s}_k)^T \hat{s}_k^p \leq 0 .$$

This implies that in all cases the lemma is true. ■

Lemma (4.9)

Let $\hat{s}_k^q$ be as in Corollary (4.3), then there exists a constant b_3 such that:

$$|| \hat{s}_k^q ||_2 \leq b_3 || h_k ||_2$$

Proof

Since

$$|| \hat{s}_k^q ||_2 = || Q_k \hat{s}_k ||_2$$

$$= || \nabla h_k (\nabla h_k^T \nabla h_k)^{-1} \nabla h_k^T \hat{s}_k ||_2$$

Equivalently, we can write

$$|| \hat{s}_k^q ||_2 = || \nabla h_k (\nabla h_k^T \nabla h_k)^{-1} (h_k + \nabla h_k^T \hat{s}_k - h_k) ||_2$$

Hence,

$$|| \hat{s}_k^q ||_2 \leq || \nabla h_k (\nabla h_k^T \nabla h_k)^{-1} ||_2 [|| h_k + \nabla h_k^T \hat{s}_k - h_k ||_2]$$

Now, from the definition of $\hat{s}_k$, we can write

$$|| \hat{s}_k^q ||_2 \leq 2 || \nabla h_k (\nabla h_k^T \nabla h_k)^{-1} ||_2 || h_k ||_2 \qquad (4.2.8)$$

Set

$$b_3 = 2 \sup_{x \in \Omega} || \nabla h(x) (\nabla h(x)^T \nabla h(x))^{-1} ||_2$$

The result now follows if we substitute b_3 in (4.2.8). ■

Lemma (4.10)

Let $\hat{s}_k$ be the step generated by the algorithm. Let P_k , $\overline{\Delta}_k$, $\hat{s}_k^p$, $\hat{s}_k^q$ be as in Corollary (4.3) and $\bar{h}_k = \nabla h_k (\nabla h_k^T \nabla h_k)^{-1} h_k$, then

$$Pred_k \geq \frac{1}{4} || P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2 \min [\overline{\Delta}_k , \frac{|| P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2}{2 b_0}]$$

$$- b_4 || \hat{s}_k ||_2 || h_k ||_2 - | (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k |$$

$$+ r_k [|| h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2] , \qquad (4.2.9)$$

where b_0 and b_4 are constants independent of k .

Proof

Since

$$Pred_k = -\nabla l_k^T \hat{s}_k - \frac{1}{2}\hat{s}_k^T B_k \hat{s}_k - \hat{\Delta\lambda}_k^T (h_k + \nabla h_k^T \hat{s}_k) + r_k [\| h_k \|^2 - \| h_k + \nabla h_k^T \hat{s}_k \|_2^2],$$

we can write

$$Pred_k = -(\nabla l_k + B_k \hat{s}_k)^T \hat{s}_k + \frac{1}{2}\hat{s}_k^T B_k \hat{s}_k + (\nabla l_k + B_k \hat{s}_k)^T \nabla h_k (\nabla h_k^T \nabla h_k)^{-1}(h_k + \nabla h_k^T \hat{s}_k) + r_k [\| h_k \|_2^2 - \| h_k + \nabla h_k^T \hat{s}_k \|_2^2].$$

Now, since $\nabla h_k (\nabla h_k^T \nabla h_k)^{-1} \nabla h_k^T \hat{s}_k = \hat{s}_k^q$, we can write:

$$Pred_k = -(\nabla l_k + B_k \hat{s}_k)^T \hat{s}_k + \frac{1}{2}\hat{s}_k^T B_k \hat{s}_k + (\nabla l_k + B_k \hat{s}_k)^T [\bar{h}_k + \hat{s}_k^q] + r_k [\| h_k \|_2^2 - \| h_k + \nabla h_k^T \hat{s}_k \|_2^2].$$

Since $\hat{s}_k - \hat{s}_k^q = \hat{s}_k^p$, we get

$$Pred_k = -(\nabla l_k + B_k \hat{s}_k)^T \hat{s}_k^p + \frac{1}{2}\hat{s}_k^T B_k \hat{s}_k + (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k + r_k [\| h_k \|_2^2 - \| h_k + \nabla h_k^T \hat{s}_k \|_2^2].$$

But by using Lemma (4.8), we can write

$$-(\nabla l_k + B_k \hat{s}_k)^T \hat{s}_k^p \geq -\frac{1}{2}(\nabla l_k + B_k \hat{s}_k)^T \hat{s}_k^p . \qquad (4.2.10)$$

Now

$$Pred_k \geq -\frac{1}{2}(\nabla l_k + B_k \hat{s}_k)^T \hat{s}_k^p + \frac{1}{2}\hat{s}_k^T B_k \hat{s}_k + (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k + r_k [\| h_k \|_2^2 - \| h_k + \nabla h_k^T \hat{s}_k \|_2^2]$$

$$\geq -\frac{1}{2}(\nabla l_k + B_k \hat{s}_k^q)^T \hat{s}_k^p - \frac{1}{2}(\hat{s}_k^p)^T B_k \hat{s}_k^p + \frac{1}{2}\hat{s}_k^T B_k \hat{s}_k$$

$$+ (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k + r_k [\; || h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2] ;$$

which can be written as

$$Pred_k \geq -\frac{1}{2} (\nabla l_k + B_k \hat{s}_k^q)^T \hat{s}_k^p + \frac{1}{2} (\hat{s}_k^p)^T B_k \hat{s}_k^q + \frac{1}{2} \hat{s}_k^T B_k \hat{s}_k^q$$
$$+ (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k + r_k [\; || h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2] .$$

By using Corollary (4.4), we get

$$Pred_k \geq \frac{1}{4} || P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2 \min [\bar{\Delta}_k , \frac{|| P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2}{2 || B_k ||_2}]$$
$$+ \frac{1}{2} (\hat{s}_k^p)^T B_k \hat{s}_k^q + \frac{1}{2} \hat{s}_k^T B_k \hat{s}_k^q + (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k$$
$$+ r_k [\; || h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2].$$

But by lemma (4.8), $|| \hat{s}_k^q ||_2 \leq b_3 || h_k ||_2$, $|| \hat{s}_k^p ||_2 \leq || \hat{s}_k ||_2$, and from the standard assumptions there exists a constant b_0 such that $|| B_k ||_2 \leq b_0$. So, we can write

$$Pred_k \geq \frac{1}{4} || P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2 \min [\bar{\Delta}_k , \frac{|| P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2}{2 b_0}]$$
$$- (b_0 b_3 || \hat{s}_k ||_2 || h_k ||_2) - | (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k |$$
$$+ r_k [\; || h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2].$$

If we set $b_4 = b_0 \, b_3$, we will get the result. ∎

The first term and the fourth term in (4.2.9) are positive, and the second and the third are negative. In order to prove that we will get a positive predicted reduction each iteration, we have to prove that the positive quantities are greater than or equal to the negative quantities otherwise we have to increase the penalty

parameter to insure that. First we need to get an upper bound on the third quantity: Corollary (4.12) will give us that bound. But first we need the following lemma

Lemma (4.11)

Let Q_k be as in Corollary (4.3), then there exist constants b_5 and b_6 such that

$$|| \, Q_k(\nabla l_k + B_k \hat{s}_k) \, ||_2 \leq b_5 \, || \, \hat{s}_k \, ||_2 + b_6 \, || \, s_{k-t_k} \, ||$$

where s_{k-t_k} is the last acceptable step and $k - t_k \geq 0$.

Proof

We have

$$Q_k(\nabla l_k + B_k \, \hat{s}_k) = Q_k \, \nabla f_k + Q_k \, \nabla h_k \, \lambda_k + Q_k \, B_k \, \hat{s}_k$$

Now, since

$$Q_k \nabla f_k = \nabla h_k \, (\, \nabla h_k^T \, \nabla h_k \,)^{-1} \, \nabla h_k^T \nabla f_k = - \nabla h_k \, \lambda_k^p$$

where $\lambda_k^p = - (\, \nabla h_k^T \, \nabla h_k \,) \, \nabla h_k^T \, \nabla f_k$, and since ,

$$Q_k \, \nabla h_k = \nabla h_k \, (\, \nabla h_k^T \, \nabla h_k \,)^{-1} \, \nabla h_k^T \, \nabla h_k = \nabla h_k \, ,$$

we have

$$Q_k \, \nabla h_k \, \lambda_k = \nabla h_k \, \lambda_k = \nabla h_k \, \lambda_{k-t_k+1}$$

$$= - \nabla h_k \, [\, (\, \nabla h_{k-t_k}^T \, \nabla h_{k-t_k} \,)^{-1} \nabla h_{k-t_k}^T \, (\, \nabla f_{k-t_k} + B_{k-t_k} \, s_{k-t_k} \,) \,]$$

$$= \nabla h_k \, [\, \lambda_{k-t_k}^p - (\, \nabla h_{k-t_k}^T \, \nabla h_{k-t_k} \,)^{-1} \nabla h_{k-t_k}^T \, B_{k-t_k} \, s_{k-t_k} \,] \, .$$

This implies that

$$\begin{aligned} \| Q_k(\nabla l_k + B_k \hat{s}_k) \|_2 \leq & \| \nabla h_k (\lambda_k^p - \lambda_{k-t_k}^p) \|_2 \\ & + b_1 \| \nabla h_k \|_2 \| B_{k-t_k} \|_2 \| s_{k-t_k} \|_2 \\ & + \| B_k \|_2 \| \hat{s}_k \|_2 \end{aligned} \tag{4.2.11}$$

Now by using the standard assumptions, there exists a constant b_7, such that

$$\begin{aligned} \| \nabla h_k (\lambda_k^p - \lambda_{k-t_k}^p) \|_2 \leq & \| \nabla h_k \|_2 \| \lambda_k^p - \lambda_{k-t_k}^p \|_2 \\ \leq & b_7 \| x_k - x_{k-t_k} \|_2 , \end{aligned}$$

and since $x_k = x_{k-t_k+1}$, we have

$$\begin{aligned} \| \nabla h_k (\lambda_k^p - \lambda_{k-t_k}^p) \|_2 \leq & b_7 \| x_{k-t_k+1} - x_{k-t_k} \|_2 \\ \leq & b_7 \| s_{k-t_k} \|_2 . \end{aligned} \tag{4.2.12}$$

Substitute (4.2.12) in (4.2.11), and by using the standard assumption, we obtain

$$\| Q_k(\nabla l_k + B_k \hat{s}_k) \|_2 \leq b_5 \| \hat{s}_k \|_2 + b_6 \| s_{k-t_k} \|_2 .$$

Hence we get the desired result. ∎

Corollary (4.12)

Let $\bar{h}_k$ be as in Lemma (4.10), then there exist constants a_4 and a_5 such that

$$| (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k | \leq [a_4 \| \hat{s}_k \|_2 + a_5 \| s_{k-t_k} \|_2] \| h_k \|_2$$

where s_{k-t_k} is the last acceptable step and $k - t_k \geq 0$.

Proof

Since $Q_k \bar{h}_k = \bar{h}_k$, we have

$$| (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k | = | [Q_k (\nabla l_k + B_k \hat{s}_k)]^T \bar{h}_k |$$

$$\leq \; || \, Q_k \, (\nabla l_k + B_k \hat{s}_k) \, ||_2 \; || \, \bar{h}_k \, ||_2$$

Now, by using Lemma (4.11) and the fact that $|| \, \bar{h}_k \, ||_2 \leq b_8 \; || \, h_k \, ||_2$, where $b_8 = \sup_{x \, \epsilon \, \Omega} || \, \nabla h(x)(\nabla h(x)^T \nabla h(x))^{-1} \, ||_2$ the proof follows immediately. ■

The following lemma proves that if $|| \, h_k \, ||_2$ is small enough, then we do not need to increase the penalty parameter in step (3) of the algorithm.

Lemma (4.13)

Let k indexed an iteration at which the algorithm does not terminate, if $|| \, h_k \, ||_2 \leq c_2 \, \Delta_k$ where c_2 is a small constant that satisfies

$$c_2 \leq \min \left[\frac{\sqrt{3}}{2 \, b_3} , \frac{\epsilon}{3 \, \Delta_*} , \frac{\epsilon}{3 \, b_4 \, \Delta_*} , \frac{\epsilon}{48 \, (a_4 + b_4 + a_5) \, \Delta_*} \min \left(1 , \frac{\epsilon}{3 \, b_0 \, \Delta_*} \right) \right] . \quad (4.2.13)$$

where a_4 and a_5 are as in Lemma (4.12), b_3 is as in Lemma (4.9), b_4 is as in Lemma (4.10), and Δ_* is an upper bound on the trust region radius, then

$$Pred_k \; \geq \; \frac{r_k}{2} \left[\; || \, h_k \, ||_2^2 - \; || \, h_k + \nabla h_k^T \hat{s}_k \, ||_2^2 \right]$$

$$+ \frac{1}{8} \; || \, P_k(\nabla l_k + B_k \; \hat{s}_k^q) \, ||_2 \min \left[\frac{1}{2} \Delta_k , \frac{|| \, P_k(\nabla l_k + B_k \; \hat{s}_k^q) \, ||_2}{2 b_0} \right] .$$

Proof

If k is the index of an iteration at which the algorithm does not terminate, then

$$|| \, P_k \, \nabla l_k \, ||_2 + \; || \, h_k \, ||_2 \; \geq \; \epsilon \, .$$

But, since $|| \, h_k \, ||_2 \leq \frac{1}{3} \, \epsilon$, it follows that

$$|| \, P_k \, \nabla l_k \, ||_2 \; \geq \; \frac{2}{3} \, \epsilon$$

Now

$$\begin{aligned} || P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2 &\geq || P_k \nabla l_k ||_2 - || P_k B_k \hat{s}_k^q ||_2 \\ &\geq || P_k \nabla l_k ||_2 - || B_k ||_2 || \hat{s}_k^q ||_2 \\ &\geq || P_k \nabla l_k ||_2 - b_0 b_3 || h_k ||_2 \\ &= || P_k \nabla l_k ||_2 - b_4 || h_k ||_2 \qquad (4.2.14) \\ &\geq \frac{2}{3}\epsilon - \frac{1}{3}\epsilon . \end{aligned}$$

Hence,

$$|| P_k (\nabla l_k + B_k \hat{s}_k^q) ||_2 \geq \frac{1}{3}\epsilon . \qquad (4.2.15)$$

Now, from Lemma (4.10), Corollary (4.12) and $|| h_k ||_2 \leq c_2 \Delta_k$, we get

$$\begin{aligned} Pred_k \geq\ & \frac{1}{4} || P_k (\nabla l_k + B_k \hat{s}_k^q) ||_2 \min [\overline{\Delta}_k , \frac{|| P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2}{2 b_0}] \\ & - c_2 [b_4 || \hat{s}_k ||_2 - (a_4 || \hat{s}_k ||_2 + a_5 || s_{k-t_k} ||_2)] \Delta_k \\ & + r_k [|| h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2] . \qquad (4.2.16) \end{aligned}$$

So, by using (4.2.15), we can write

$$\begin{aligned} Pred_k \geq\ & \frac{1}{8} || P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2 \min [\overline{\Delta}_k , \frac{|| P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2}{2 b_0}] \\ & + \frac{1}{8} (\frac{1}{3}\epsilon) \min [\overline{\Delta}_k , \frac{\epsilon}{6 b_0}] - c_2 [(a_4 + b_4 + a_5) \Delta_*] \Delta_k \\ & + r_k [|| h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2] . \qquad (4.2.17) \end{aligned}$$

Now, since

$$\overline{\Delta}_k = \sqrt{\Delta_k^2 - || \hat{s}_k^q ||_2^2} ,$$

by using Lemma (4.9) and $|| h_k ||_2 \leq \frac{\sqrt{3}}{2\, b_3} \Delta_k$,

$$\overline{\Delta}_k \geq \sqrt{\Delta_k^2 - b_3^2 \, || h_k ||_2^2} ,$$

and we obtain

$$\overline{\Delta}_k \geq \sqrt{\Delta_k^2 - (3/4)\, \Delta_k^2}$$

$$= \frac{1}{2} \Delta_k .$$

By substituting the last inequality in (4.2.17), we get

$$\begin{aligned} Pred_k \geq & \frac{1}{8} \, || P_k(\nabla l_k + B_k \, \hat{s}_k^q) \, ||_2 \min \left[\frac{1}{2} \Delta_k , \frac{|| P_k(\nabla l_k + B_k \, \hat{s}_k^q) \, ||_2}{2 b_0} \right] \\ & + \frac{1}{8} \left(\frac{1}{3} \epsilon \right) \min \left[\frac{1}{2} \Delta_k , \frac{\epsilon}{6\, b_0} \right] - c_2 \left[(a_4 + b_4 + a_5) \, \Delta_* \right] \Delta_k \\ & + r_k \left[\, || h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2 \, \right] . \end{aligned}$$

Since c_2 satisfies inequality (4.2.13), we have

$$\begin{aligned} Pred_k \geq & \frac{1}{8} \, || P_k(\nabla l_k + B_k \, \hat{s}_k^q) \, ||_2 \min \left[\frac{1}{2} \Delta , \frac{|| P_k(\nabla l_k + B_k \, \hat{s}_k^q) \, ||_2}{2 b_0} \right] \\ & + \frac{r_k}{2} \left[\, ||h_k ||_2^2 - ||h_k + \nabla h_k^T \hat{s}_k ||_2^2 \, \right] . \end{aligned} \tag{4.2.18}$$

Hence we get the desired result. ■

If $|| h_k ||_2 \leq c_2 \, \Delta_k$, then half of the first term in (4.2.16) would cancel the second and the third terms, and the fourth term need never enter the calculation. This implies that if we set $r_k = r_{k-1}$, inequality (4.2.18) remains correct. So, in this case, we do not need to increase the penalty parameter.

Lemma (4.14)

Let k be the index of an iteration at which the algorithm does not terminate. If $|| h_k ||_2 \leq c_2 \Delta_k$, where c_2 is as in Lemma (4.13), then there exists a constant c_3 such that

$$Pred_k \geq c_3 \Delta_k$$

Proof

From (4.2.15) and (4.2.18), we have

$$Pred_k \geq \frac{1}{8} (\frac{1}{3} \epsilon) \min [\frac{1}{2}\Delta_k , \frac{\epsilon}{6 b_0}]$$

$$\geq \frac{1}{48} \epsilon \min [1 , \frac{\epsilon}{3 b_0 \Delta_*}] \Delta_k .$$

The result now follows if we set $c_3 = \frac{1}{48} \epsilon \min [1 , \frac{\epsilon}{3 b_0 \Delta_*}]$. ∎

4.3 THE BEHAVIOR OF THE PENALTY PARAMETER

This section is devoted to the study of the behavior of the penalty parameter. The following three lemmas are needed to prove that the penalty parameter is bounded. In Lemma (4.18) we prove that the penalty parameter will remain bounded as long as the algorithm does not terminate.

Lemma (4.15)

If k is the index of an iteration at which the penalty parameter r_k increases, we have

$$r_k \min [\, \Delta_k \,,\, \frac{|| h_k ||_2}{b_2} \,] \leq a_6 \, || \hat{s}_k ||_2 + a_7 \, || s_{k-t_k} ||_2$$

where a_6 and a_7 are constants independent of k, s_{k-t_k} is the last acceptable step and $k-t_k \geq 0$.

Proof

Let k be the index of an iteration at which the penalty parameter increases, then by step 3 of the algorithm r_k is updated by the following rule:

$$r_k = 2 \, \frac{\nabla_x l_k^T \hat{s}_k + \tfrac{1}{2} \hat{s}_k^T B_k \hat{s}_k + \hat{\Delta\lambda}_k^T (h_k + \nabla h_k^T \hat{s}_k)}{|| h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2} + \rho \,.$$

This can be written as

$$\begin{aligned} \frac{r_k}{2} [\, || h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2 \,] &= (\nabla l_k + B_k \hat{s}_k)^T \hat{s}_k^p - \frac{1}{2} \hat{s}_k^T B_k \hat{s}_k \\ &\quad - (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k \\ &\quad + \frac{\rho}{2} [\, || h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2 \,] . \end{aligned}$$

Using Lemma (4.1), inequality (4.2.10) and $\hat{s}_k = \hat{s}_k^p + \hat{s}_k^q$, we get

$$\begin{aligned} \frac{r_k}{2} \frac{|| h_k ||_2}{b_1} \min [\, \Delta_k \,,\, \frac{|| h_k ||_2}{b_2} \,] &\leq \frac{1}{2} (\nabla l_k + B_k \hat{s}_k^q)^T \hat{s}_k^p + \frac{1}{2} \hat{s}_k^{pT} B_k \hat{s}_k^p \\ &\quad - \frac{1}{2} \hat{s}_k^T B_k \hat{s}_k - (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k \\ &\quad + \frac{\rho}{2} [\, || h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2 \,] . \end{aligned}$$

By using Corollary (4.4), we can write

$$\begin{aligned} &\frac{r_k}{2} \frac{|| h_k ||_2}{b_1} \min [\, \Delta_k \,,\, \frac{|| h_k ||_2}{b_2} \,] \\ &\leq -\frac{1}{2} \, || P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2 \min [\, \overline{\Delta}_k \,,\, \frac{|| P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2}{2 b_0} \,] \end{aligned}$$

$$+ \frac{1}{2} \hat{s}_k^{pT} B_k \hat{s}_k^p - \frac{1}{2} \hat{s}_k^T B_k \hat{s}_k - (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k$$

$$- \rho [h_k^T \nabla h_k^T \hat{s}_k + \frac{1}{2} \hat{s}_k^T \nabla h_k \nabla h_k^T \hat{s}_k] .$$

Thus,

$$\frac{r_k}{2} \frac{|| h_k ||_2}{b_1} \min [\Delta_k , \frac{|| h_k ||_2}{b_2}] \leq - \frac{1}{2} (\hat{s}_k^p)^T B_k \hat{s}_k^q - \frac{1}{2} \hat{s}_k^T B_k \hat{s}_k^q$$
$$- (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k$$
$$- \rho h_k^T \nabla h_k^T \hat{s}_k ,$$

and we can write

$$\frac{r_k}{2} \frac{|| h_k ||_2}{b_1} \min [\Delta_k , \frac{|| h_k ||_2}{b_2}] \leq || B_k ||_2 || \hat{s}_k ||_2 || \hat{s}_k^q ||_2$$
$$+ \rho || \nabla h_k ||_2 || \hat{s}_k ||_2 || h_k ||_2$$
$$+ | (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k | . \quad (4.3.1)$$

Now by using Corollary (4.12),

$$\frac{r_k}{2} \frac{|| h_k ||_2}{b_1} \min [\Delta_k , \frac{|| h_k ||_2}{b_2}] \leq || B_k ||_2 || \hat{s}_k ||_2 || \hat{s}_k^q ||_2$$
$$+ (a_4 || \hat{s}_k ||_2 + a_5 || s_{k-t_k} ||_2) || h_k ||_2$$
$$+ \rho || \nabla h_k ||_2 || \hat{s}_k ||_2 || h_k ||_2 .$$

But, by Lemma (4.8) $|| \hat{s}_k^q || \leq b_3 || h_k ||$ and from the standard assumptions $|| \nabla h_k ||_2 \leq b_9$ where $b_9 = \sup_{x \in \Omega} || \nabla h(x) ||$,

$$\frac{r_k || h_k ||_2}{2 b_1} \min [\Delta_k , \frac{||h_k ||_2}{b_2}] \leq (b_0 b_3 + a_4 + \rho b_9) || \hat{s}_k ||_2 || h_k ||_2$$
$$+ a_5 || s_{k-t_k} ||_2 || h_k ||_2 .$$

The result follows immediately if we divide by $\frac{|| h_k ||_2}{b_1}$. ■

Corollary (4.16)

If k is the index of an iteration at which the algorithm does not terminate and the penalty parameter r_k increases, we have

$$r_k \Delta_k \leq a_8 || \hat{s}_k ||_2 + a_9 || s_{k-t_k} ||_2$$

where a_8 and a_9 are constants independent of k, s_{k-t_k} is the last acceptable step and $k-t_k \geq 0$.

Proof

From Lemma (4.15), if k is index of an iteration at which the penalty parameter r_k increases, then r_k must satisfy the following inequality:

$$r_k \min [\Delta_k , \frac{|| h_k ||_2}{b_2}] \leq a_6 || \hat{s}_k ||_2 + a_7 || s_{k-t_k} ||_2$$

From Lemma (4.13) if $|| h_k ||_2 \leq c_2 \Delta_k$, then we do not increase r_k. So, for any iteration at which the penalty parameter increases, we must have

$$|| h_k ||_2 > c_2 \Delta_k ,$$

and we get

$$r_k \min [\Delta_k , \frac{c_2 \Delta_k}{b_2}] \leq a_6 || \hat{s}_k ||_2 + a_7 || s_{k-t_k} ||_2 .$$

This can be written as

$$r_k \Delta_k \min [1 , \frac{c_2}{b_2}] \leq a_6 || \hat{s}_k ||_2 + a_7 || s_{k-t_k} ||_2 .$$

Hence,

$$r_k \Delta_k \leq a_8 \| \hat{s}_k \|_2 + a_9 \| s_{k-t_k} \|_2 ,$$

and we get the desired result. ■

By the standard assumptions, at each iteration at which the penalty parameter increases, $r_k \Delta_k$ is bounded. However, if we can bound $\frac{\| s_{k-t_k} \|_2}{\Delta_k}$ by a constant independent of k, we can get an upper bound on r_k itself. In the following lemma we get a relation between $\| s_{k-t_k} \|_2$ and Δ_k. In Lemma (4.18) we prove that the penalty parameter is bounded.

Lemma (4.17)

Let k be the index of any iteration at which the algorithm does not terminate and the penalty parameter r_k increases, then

$$\Delta_k \geq c_4 \| s_{k-t_k} \|_2$$

where s_{k-t_k} is the last acceptable step, c_4 is a constant independent of k and t_k, and $k-t_k \geq 0$.

Proof

Let us consider three cases:

First, if $t_k = 1$, *i.e.*, s_{k-1} is the last acceptable step, then from (3.1.3), we have

$$\Delta_k \geq \alpha_1 \| s_{k-1} \|_2$$

The result in this case follows if we set $c_4 = \alpha_1$

Second, if s_{k-1} is not the last acceptable step, but $|| h_{k-i} ||_2 \geq c_2 \Delta_{k-i}$ for all $i \in [1 , t_k - 1]$. In this case, from Corollary (4.7), we have

$$| Ared_{k-i} - Pred_{k-i} | \leq a_0 r_{k-i} || \hat{s}_{k-i} ||_2^2$$

Now, from Corollary (4.2), we have

$$Pred_{k-i} \geq \frac{r_{k-i}}{2} \frac{|| h_{k-i} ||_2}{b_1} \min [\Delta_{k-i} , \frac{|| h_{k-i} ||_2}{b_2}] ,$$

But since all $k-i$, $i=1,...,t_k-1$ satisfy $|| h_{k-i} ||_2 \geq c_2 \Delta_{k-i} \geq c_2 || \hat{s}_{k-i} ||_2$, we have

$$Pred_{k-i} \geq \frac{r_{k-i}}{2} \frac{|| h_{k-i} ||_2}{b_1 b_2} || \hat{s}_{k-i} ||_2 \min [b_2 , c_2] .$$

Hence,

$$| \frac{Ared_{k-i} - Pred_{k-i}}{Pred_{k-i}} | \leq \frac{2 a_0 b_1 b_2 || \hat{s}_{k-i} ||_2}{\min [b_2 , c_2] || h_{k-i} ||_2} .$$

But since all $k-i$, $i=1,...,t_k-1$ index unacceptable steps, we have

$$(1 - \eta_1) < | \frac{Ared_{k-i}}{Pred_{k-i}} - 1 | , \qquad 1 \leq i \leq t_k - 1$$

So, for all $i \in [1 , t_k - 1]$, we have

$$|| \hat{s}_{k-i} ||_2 \geq \frac{(1 - \eta_1)}{2 a_0 b_1 b_2} \min [b_2 , c_2] || h_{k-i} ||_2 .$$

Now, since $x_{k-1} = x_{k-(t_k-1)}$, $h_{k-1} = h_{k-(t_k-1)}$, we have

$$\Delta_k \geq \alpha_1 || \hat{s}_{k-1} ||_2$$

$$\geq \frac{\alpha_1 (1 - \eta_1)}{2 a_0 b_1 b_2} \min [b_2 , c_2] || h_{k-1} ||_2$$

$$= \frac{\alpha_1 (1 - \eta_1)}{2 a_0 b_1 b_2} \min [b_2 , c_2] || h_{k-(t_k-1)} ||_2$$

$$\geq \frac{\alpha_1 \, c_2 \, (\, 1 - \eta_1 \,)}{2 \, a_0 \, b_1 \, b_2} \min \, [\, b_2 \, , \, c_2 \,] \, \Delta_{k-(t_k-1)}$$

$$\geq \frac{\alpha_1^2 \, c_2 \, (\, 1 - \eta_1 \,)}{2 \, a_0 \, b_1 \, b_2} \min \, [\, b_2 \, , \, c_2 \,] \quad || \, s_{k-t_k} \, ||_2 \, .$$

The result in this case follows by setting

$$c_4 \; = \; \frac{\alpha_1^2 \, c_2 \, (\, 1 - \eta_1 \,)}{2 \, a_0 \, b_1 \, b_2} \min \, [\, b_2 \, , \, c_2 \,] \, .$$

Finally, if the step indexed by $k-1$ is not the last acceptable step and not all $i \in [\, 1 \, , \, t_k - 1 \,]$ satisfy $|| \, h_{k-i} \, ||_2 \geq c_2 \, \Delta_{k-i}$, then there exists at least one $j \in [\, 1 \, , \, t_k - 1 \,]$ such that $|| \, h_{k-j} \, ||_2 < c_2 \, \Delta_{k-j}$. Let l be the smallest integer $\in [\, 1 \, , \, t_k - 1 \,]$ such that $|| \, h_{k-l} \, ||_2 < c_2 \, \Delta_{k-l}$. For all $i \in [\, 1 \, , \, l-1 \,]$, we have

$$|| \, h_{k-i} \, ||_2 \geq c_2 \, \Delta_{k-i}$$

As in the first two parts, if we set

$$c_5 \; = \; \min \, [\, \alpha_1 \, , \, \frac{\alpha_1^2 \, c_2 \, (\, 1 - \eta_1 \,)}{2 \, a_0 \, b_1 \, b_2} \min \, (\, b_2 \, , \, c_2 \,) \,] \, , \tag{4.3.2}$$

we get

$$\Delta_k \; \geq \; c_5 \, || \, \hat{s}_{k-l} \, ||_2 \tag{4.3.3}$$

where c_5 is given by (4.3.2). Now, for $k-l$ we have

$$|| \, h_{k-l} \, ||_2 \; < \; c_2 \, \Delta_{k-l} \, . \tag{4.3.4}$$

From Lemma (4.6) we have

$$| \, Ared_{k-l} - Pred_{k-l} \, | \leq a_1 \, || \, \hat{s}_{k-l} \, ||_2^2 + r_{k-l} [\, a_2 \, || \, \hat{s}_{k-l} \, ||_2^3 + a_3 \, || \, h_{k-l} \, ||_2 \, || \, \hat{s}_{k-l} \, ||_2^2 \,] \, .$$

By using inequality (4.3.4), we have

$$| \, Ared_{k-l} - Pred_{k-l} \, | \; \leq \; a_1 \, || \, \hat{s}_{k-l} \, ||_2^2 + r_{k-l} \, (\, a_2 + a_3 \, c_2 \,) \, || \, \hat{s}_{k-l} \, ||_2^2 \, \Delta_{k-l} \, . \tag{4.3.5}$$

If k indexes an iteration at which r_k increases, then from Lemma (4.16) and the standard assumptions we know that $r_k\ \Delta_k$ is bounded. By using inequality (4.3.3), we get

$$\begin{aligned} r_{k-l}\ \|\ \hat{s}_{k-l}\ \|_2 &\leq \frac{1}{c_5}\, r_{k-l}\ \Delta_k \\ &\leq \frac{1}{c_5}\, r_k\ \Delta_k \ \leq\ m_0\,, \end{aligned}$$

where m_0 is a uniform bound.

Hence inequality (4.3.5) can be written as

$$\begin{aligned} |\ Ared_{k-l} - Pred_{k-l}\ | &\leq a_1\ \|\ \hat{s}_{k-l}\ \|_2^2 + (\ a_2 + c_2\ a_3\ )\ m_0\ \|\ \hat{s}_{k-l}\ \|_2\ \Delta_{k-l} \\ &\leq [\ a_1 + (\ a_2 + c_2\ a_3\ )\ m_0\ ]\ \|\ \hat{s}_{k-l}\ \|_2\ \Delta_{k-l}\ . \end{aligned}$$

By using Lemma (4.14), we get

$$\begin{aligned} \left|\ \frac{Ared_{k-l} - Pred_{k-l}}{Pred_{k-l}}\ \right| &\leq \frac{[\ a_1 + (\ a_2 + c_2\ a_3\ )\ m_0\ ]\ \|\ \hat{s}_{k-l}\ \|_2\ \Delta_{k-l}}{c_3\ \Delta_{k-l}} \\ &= \frac{a_1 + (\ a_2 + c_2\ a_3\ )\ m_0}{c_3}\ \|\ \hat{s}_{k-l}\ \|_2\,. \end{aligned}$$

But since the $k-l^{\text{th}}$ is not an acceptable step, then

$$\begin{aligned} (\ 1 - \eta_1\ ) &< \left|\ \frac{Ared_{k-l}}{Pred_{k-l}} - 1\ \right| \\ &\leq \frac{a_1 + (\ a_2 + c_2\ a_3\ )\ m_0}{c_3}\ \|\ \hat{s}_{k-l}\ \|_2\,. \end{aligned}$$

Hence, by using inequality (4.3.3), we obtain

$$\begin{aligned} \Delta_k &\geq c_5\ \|\ \hat{s}_{k-l}\ \|_2 \\ &> \frac{c_3\ c_5}{[\ a_1 + (\ a_2 + c_2\ a_3\ )\ m_0\ ]}\ (\ 1 - \eta_1\ )\,. \end{aligned}$$

$$\geq \frac{c_3\, c_5\, (\, 1 - \eta_1\,)}{[\, a_1 + (\, a_2 + c_2\, a_3\,)\, m_0\,]\, \Delta_*} \;||\; s_{k-t_k} \;||_2$$

The result then follows if we set

$$c_4 \;=\; \min\, [\; c_5\;,\; \frac{c_3\, c_5\, (\, 1 - \eta_1\,)}{[\, a_1 + (\, a_2 + c_2\, a_3\,)\, m_0\,]\, \Delta_*}\;] \;.$$

This completes the proof. ■

Lemma (4.18)

Under the standard assumptions, if the algorithm does not terminate, the penalty parameter r_k is bounded.

Proof

The proof is by contradiction. Suppose that r_k is not bounded. This implies that there exists an infinite subsequence of indices $\{k_j\}$ at which $\{r_{k_j}\}$ is unbounded. Now, from Lemma (4.13), we never increase the penalty parameter if $||\; h_k \;||_2 \leq c_2\, \Delta_k$. So, $||\; h_{k_j} \;||_2 > c_2\, \Delta_{k_j}$.

Let m be any integer $\epsilon\, \{\, k_j \,\}$, then from Corollary (4.16) we can write

$$r_m \;\; \Delta_m \;\leq\; a_8 \;||\; \hat{s}_m \;||_2 + a_9 \;||\; s_{m-t_m} \;||_2 \;, \qquad (4.3.6)$$

where s_{m-t_m} is the last acceptable step. On the other hand, from Lemma (4.17) we have

$$\Delta_m \;\geq\; c_4 \;\; ||\; s_{m-t_m} \;||_2 \;.$$

Hence

$$||\; s_{m-t_m} \;||_2 \;\leq\; \frac{1}{c_4}\, \Delta_m \;.$$

By substituting the last inequality in (4.3.6), we get

$$r_m \leq a_8 + \frac{a_9}{c_4} .$$

Set

$$N = (a_8 + \frac{a_9}{c_4}) .$$

Since N is independent of m , it is an upper bound of the sequence $\{ r_{k_j} \}$ contradicting the assumption that the sequence $\{ r_{k_j} \}$ has no upper bound. This proves the theorem. ■

From the last lemma, we can conclude that for all k , $1 \leq r_k \leq r_*$ where r_* is a constant independent of k .

Since if r_k increases, it will increase by a quantity $\geq \rho$, then the number of iterations at which the penalty parameter increases is finite. Hence, there exists a constant $\bar{k}$ such that $r_k = r_{\bar{k}}$ for all $k \geq \bar{k}$.

4.4 THE GLOBAL CONVERGENCE THEORY

In this part we present our global convergence theory. We start by proving that the algorithm is well defined in the sense that it always finds an acceptable step from any point that does not satisfy the termination criteria. Then we prove that the algorithm will terminate at a point within ϵ of a Kuhn-Tucker point.

We call an iteration a successful iteration if the trial step of that iteration is accepted because $\frac{Ared_k}{Pred_k} > \eta_1$. Otherwise, the iteration is said to be unsuccess-

ful.

We denote by $S(k_1, k_2)$ the set of indices of successful iterations in the interval $[k_1, k_2]$.

The following theorem shows that the algorithm is well defined in the sense that at any iteration either the point (x_k, λ_k) is within ϵ of a Kuhn-Tucker point and the termination condition of the algorithm will be met or the algorithm will always find an acceptable step.

Theorem (4.19)

Under the standard assumptions, either the point (x_k, λ_k) is within ϵ of a Kuhn-Tucher point and the termination condition of the algorithm will be met or we always find an acceptable step. *i.e.* the condition $\frac{Ared_{k+j}}{Pred_{k+j}} \geq \eta_1$ will be satisfied for some j.

Proof

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 in step 1 of the algorithm.

First, we assume that $||h_k||_2 > c_2 \Delta_k$ where c_2 is as in Lemma (4.13). Since, from Corollary (4.2), we have

$$Pred_k \geq \frac{r_k}{2} \frac{||h_k||_2}{b_1} \min\left[\Delta_k, \frac{||h_k||_2}{b_2}\right],$$

$$\geq \frac{r_k}{2} \frac{||h_k||_2 \Delta_k}{b_1 b_2} \min[b_2, c_2],$$

and since from Corollary (4.7),

$$| Ared_k - Pred_k | \leq a_0 r_k \Delta_k^2 , \tag{4.4.1}$$

then, we have

$$| \frac{Ared_k - Pred_k}{Pred_k} | \leq \frac{2 a_0 b_1 b_2 \Delta_k}{|| h_k ||_2 \min [b_2 , c_2]} .$$

That is,

$$| \frac{Ared_k}{Pred_k} - 1 | \leq \frac{2 a_0 b_1 b_2}{|| h_k ||_2 \min [b_2 , c_2]} \Delta_k .$$

Now, as Δ_k gets smaller, the quantity $| \frac{Ared_k}{Pred_k} - 1 |$ approaches 0 and hence the condition $\frac{Ared_k}{Pred_k} \geq \eta_1$ will be met after a finite number of trials.

Now, assume that $|| h_k ||_2 \leq c_2 \Delta_k$, from Lemma (4.14) we have

$$Pred_k \geq c_3 \Delta_k$$

This gives, using (4.4.1), that

$$| \frac{Ared_k - Pred_k}{Pred_k} | \leq \frac{a_0 r_*}{c_3} \Delta_k$$

So, as Δ_k gets smaller, the quantity $| \frac{Ared_k}{Pred_k} - 1 |$ approaches 0, and hence the condition $\frac{Ared_k}{Pred_k} \geq \eta_1$ will be met after a finite number of trials. This completes the proof. ■

The following theorem proves that under the standard Assumptions, either the algorithm terminates, or converges to a feasible point.

Theorem (4.20)

Let the standard assumptions hold. Assume that $\{ \Phi_k \}$ is bounded below on Ω. If the algorithm does not terminate, then

$$\lim_{k \to \infty} \; || \, h_k \, ||_2 \; = \; 0 \, .$$

Proof

Suppose $\limsup_{k \to \infty} \; || \, h_k \, ||_2 \; = \; \epsilon_0 \; > \; 0$. Then there exists an infinite sequence of indices $\{ k_j \}$ such that $|| \, h_k \, ||_2 \geq \frac{\epsilon_0}{2}$ for all $k \in \{ k_j \}$.

Let $\hat{k}$ be such that $\hat{k} \in \{ k_j \}$, $\hat{k} \geq \bar{k}$ and $\Delta_{\hat{k}} > 0$. Since $h \in C^2$, we have for some $\beta > 0$ and any $x \in \Omega$ that

$$\begin{aligned} || \, h(x) \, ||_2 \; &\geq \; || \, h_{\hat{k}} \, ||_2 - \; || \, h(x) - h_{\hat{k}} \, ||_2 \\ &\geq \; || \, h_{\hat{k}} \, ||_2 - \beta \; || \, x - x_{\hat{k}} \, ||_2 \end{aligned}$$

This implies that for all x that satisfies $|| \, x - x_{\hat{k}} \, ||_2 \leq \frac{|| \, h_{\hat{k}} \, ||_2}{2\beta}$, we have

$$|| \, h(x) \, ||_2 \; \geq \; \frac{|| \, h_{\hat{k}} \, ||_2}{2} \, .$$

Let $\sigma = \frac{|| \, h_{\hat{k}} \, ||_2}{2 \, \beta}$ and consider the ball

$$B_\sigma \; = \; [\, x : \; || \, x - x_{\hat{k}} \, ||_2 \leq \sigma \,] \, .$$

First we will show that eventually the iterate must move outside B_σ.

If $x_k \in B_\sigma$ for all $k \geq \hat{k}$, then from lemma (4.2) and $r_k \geq 1$,

$$Pred_k \; \geq \; \frac{1}{2} \, \frac{|| \, h_k \, ||_2}{b_1} \min \, [\, \Delta_k \, , \, \frac{|| \, h_k \, ||_2}{b_2} \,]$$

$$\geq \frac{1}{2} \frac{|| h_{\hat{k}} ||_2}{2 b_1} \min [\Delta_k , \frac{|| h_{\hat{k}} ||_2}{2 b_2}] .$$

If all $k \geq \hat{k}$ are not acceptable steps, then we get a contradiction with Theorem (4.19). Hence, there exists an infinite sequence of indices indexed successful steps inside the ball. For any such k we have

$$\begin{aligned} \Phi_k - \Phi_{k+1} &= Ared_k \\ &\geq \eta_1 \, Pred_k \\ &\geq \frac{\eta_1}{2} \frac{|| h_{\hat{k}} ||_2}{2 b_1} \min [\Delta_k , \frac{|| h_{\hat{k}} ||_2}{2 b_2}] . \end{aligned} \tag{4.4.2}$$

Since Φ_k is bounded below and $|| h_{\hat{k}} ||_2 > 0$, then inequality (4.4.2) implies that

$$\liminf_{k \to \infty} \Delta_k = 0 \tag{4.4.3}$$

Define σ_1 to be a constant that satisfies:

$$\sigma_1 < \min [1 , \frac{a \, b \, \Delta_{\hat{k}}}{\alpha_1 \, r_* (1 - \eta_2)} , \frac{|| h_{\hat{k}} ||_2}{2}]$$

where $a = \max [r_* , 2 r_*^2 a_0]$ and $b = \max [b_1 , b_2]$. Now, because of (4.4.3), there exist some sufficiently large k such that

$$\Delta_k \leq \frac{\alpha_1 \sigma_1 r_*}{a \, b} (1 - \eta_2) . \tag{4.4.4}$$

Let m be the first integer greater than $\hat{k}$ such that (4.4.4) holds. This implies that $m \geq \hat{k}+1$, and using (3.1.3) we get

$$b \, || \hat{s}_{m-1} ||_2 \leq \frac{b \, \Delta_m}{\alpha_1}$$

$$\leq \frac{\sigma_1 \, r_*}{a} (1 - \eta_2) \tag{4.4.5}$$

$$\leq \sigma_1 (1 - \eta_2) \leq \sigma_1 . \tag{4.4.6}$$

Now, by Lemma (4.2)

$$Pred_{m-1} \geq \frac{1}{2} \frac{|| h_{m-1} ||_2}{b_1} \min [|| \hat{s}_{m-1} ||_2 , \frac{|| h_{m-1} ||_2}{b_2}] , \tag{4.4.7}$$

and since $m-1 \geq \hat{k}$, we have

$$|| h_{m-1} ||_2 \geq \frac{|| h_{\hat{k}} ||_2}{2} \geq \sigma_1 . \tag{4.4.8}$$

From (4.4.6) and (4.4.8) we have

$$b \; || \hat{s}_{m-1} ||_2 \leq || h_{m-1} ||_2 .$$

By substituting the last inequality and (4.4.8) into (4.4.7), we obtain

$$Pred_{m-1} \geq \frac{\sigma_1}{2 \, b} || \hat{s}_{m-1} ||_2 .$$

But, by Corollary (4.7),

$$| Ared_{m-1} - Pred_{m-1} | \leq a_0 \, r_* \; || \hat{s}_{m-1} ||_2^2 . \tag{4.4.9}$$

So,

$$| \frac{Ared_{m-1} - Pred_{m-1}}{Pred_{m-1}} | \leq \frac{2 \, a_0 \, b \, r_* \; || \hat{s}_{m-1} ||_2^2}{\sigma_1 \; || \hat{s}_{m-1} ||_2} .$$

Now using (4.4.5), we obtain

$$| \frac{Ared_{m-1} - Pred_{m-1}}{Pred_{m-1}} | \leq \frac{2 \, a_0 \, r_*^2 \, \sigma_1}{\sigma_1 \, a} (1 - \eta_2)$$

$$\leq (1 - \eta_2) .$$

This implies that

$$\frac{Ared_{m-1}}{Pred_{m-1}} \geq \eta_2 .$$

Hence from the rule of updating the radius of the trust region, we have

$$\Delta_{m-1} \leq \Delta_m$$

The last inequality implies that $k = m-1$ satisfies (4.4.4). This contradicts the supposition that m is the smallest such index and means that there is no $m > \hat{k}$ such that (4.4.4) holds. Hence, for all $k > \hat{k}$, we have

$$\Delta_k > \frac{\alpha_1 \sigma_1 r_*}{a\ b} (1 - \eta_2)$$

which contradict (4.4.3). Hence, eventually $\{ x_k \}$ must leave the ball B_σ for some $k > \hat{k}$.

Let $l+1$ be the first integer greater than $\hat{k}$ such that x_{l+1} does not lie inside the ball B_σ. Since $x_{l+1} \neq x_{\hat{k}}$, there must exist at least one acceptable step in the set of iterates indexed $\{ \hat{k},...,l \}$, so by Lemma (4.2),

$$\Phi_{\hat{k}} - \Phi_{l+1} = \sum_{k=\hat{k}}^{l} (\Phi_k - \Phi_{k+1})$$

$$\geq \sum_{k \epsilon S(\hat{k},l)} \eta_1 Pred_k$$

$$\geq \sum_{k \epsilon S(\hat{k},l)} \frac{\eta_1}{2} \frac{|| h_{\hat{k}} ||_2}{2 b_1} \min [\Delta_k , \frac{|| h_{\hat{k}} ||_2}{2\ b_2}] .$$

If $\Delta_k < \frac{|| h_{\hat{k}} ||_2}{2\ b_2}$ for all $k \epsilon S(\hat{k},l)$, then

$$\Phi_{\hat{k}} - \Phi_{l+1} \geq \frac{\eta_1}{2} \frac{|| h_{\hat{k}} ||_2}{2\ b_1} \sum_{k \epsilon S(\hat{k},l)} \Delta_k$$

$$\geq \frac{\eta_1}{2} \frac{|| h_{\hat{k}} ||_2}{2 b_1} \sigma .$$

Otherwise,

$$\Phi_{\hat{k}} - \Phi_{l+1} \geq \frac{\eta_1}{2} \frac{|| h_{\hat{k}} ||_2^2}{4 b_1 b_2} .$$

In either case

$$\Phi_{\hat{k}} - \Phi_{l+1} \geq \frac{\eta_1}{2} \frac{|| h_{\hat{k}} ||_2}{2 b_1} \min [\sigma , \frac{|| h_{\hat{k}} ||_2}{2 b_2}]$$

$$= \frac{\eta_1}{2} \frac{|| h_{\hat{k}} ||_2}{2 b_1} \min [\frac{|| h_{\hat{k}} ||_2}{2 \beta} , \frac{|| h_{\hat{k}} ||_2}{2 b_2}]$$

$$= \frac{\eta_1}{2} \frac{|| h_{\hat{k}} ||_2^2}{4 b_1} \min [\frac{1}{\beta} , \frac{1}{b_2}] . \qquad (4.4.10)$$

Since $\{ \Phi_k \}$ is bounded below and a decreasing sequence, $\{ \Phi_k \}$ converges to some limit Φ_* . Take the limit as l goes to infinity on inequality (4.4.10), we get

$$\Phi_{\hat{k}} - \Phi_* \geq \frac{\eta_1}{2} \frac{|| h_{\hat{k}} ||_2^2}{4 b_1} \min [\frac{1}{\beta} , \frac{1}{b_2}] .$$

If we take the limit as $\hat{k}$ goes to infinity, we get

$$0 \geq \frac{\eta_1}{2} \frac{\epsilon_0}{8 b_1} \min [\frac{1}{\beta} , \frac{1}{b_2}]$$

which contradicts $\epsilon_0 > 0$. The supposition is wrong and hence the theorem is proven. ■

Theorem (4.21)

Let the standard assumptions hold. Assume that $\{ \Phi_k \}$ is bounded below on Ω .

If the algorithm does not terminate, we have

$$\liminf_{k\to\infty} \; || P_k \, \nabla l_k \, ||_2 \; = \; 0$$

Proof

The proof is by contradiction. Suppose that there exists an $\epsilon_0 > 0$ and an integer K such that $|| P_k \, \nabla l_k \, ||_2 \geq \epsilon_0$ for all $k \geq K$.

Since, by using (4.2.14),

$$|| P_k \, (\nabla l_k + B_k \, \hat{s}_k^q) \, ||_2 \geq || P_k \, \nabla l_k \, ||_2 - b_4 \, || h_k \, ||_2 ,$$

and since

$$\lim_{k\to\infty} \; || h_k \, ||_2 \; = \; 0 ,$$

there exist k_1 sufficiently large such that for all $k \geq k_1$, we have

$$|| h_k \, ||_2 \; < \; \frac{1}{2 \, b_4} \, \epsilon_0 .$$

Thus for $k \geq \max [K , k_1]$

$$|| P_k \, (\nabla l_k + B_k \, \hat{s}_k^q) \, ||_2 \geq \epsilon_0 - \frac{1}{2} \, \epsilon_0 \; = \; \frac{1}{2} \, \epsilon_0 .$$

Now, since from (4.2.9) and Corollary (4.12),

$$Pred_k \geq \frac{1}{4} \, || P_k \, (\nabla l_k + B_k \, \hat{s}_k^q) \, ||_2 \min \left[\overline{\Delta}_k , \frac{|| P_k (\nabla l_k + B_k \hat{s}_k^q) \, ||_2}{2 b_0} \right]$$

$$- (b_4 \, || \hat{s}_k \, ||_2 \, || h_k \, ||_2) - (a_4 \, || \hat{s}_k \, ||_2 + a_5 \, || s_{k-t_k} \, ||_2) \, || h_k \, ||_2 ,$$

and since $|| h_k \, ||_2$ converges to zero and $|| \hat{s}_k \, ||_2$ and $|| s_{k-t_k} \, ||_2$ are bounded, then there exists an integer $k_2 \geq \max [K , k_1]$ such that for all $k \geq k_2$ we have

$$Pred_k \geq \frac{1}{8} \| P_k(\nabla l_k + B_k \hat{s}_k^q) \|_2 \min [\frac{1}{2} \Delta_k , \frac{\| P_k(\nabla l_k + B_k \hat{s}_k^q) \|_2}{2b_0}] .$$

Thus, for all $k \geq k_2$, we have

$$Pred_k \geq \frac{1}{8} \frac{\epsilon_0}{2} \min [\frac{1}{2} \Delta_k , \frac{\epsilon_0}{4b_0}] .$$

From Theorem (4.19) there exists an infinite sequence of successful iterations. Now, for any successful iteration indexed $k \geq k_2$, we have

$$Ared_k \geq \eta_1 Pred_k$$

$$\geq \frac{\eta_1}{32} \epsilon_0 \min [\Delta_k , \frac{\epsilon_0}{2b_0}]$$

If $\bar{k}_2 \geq \max [k_2 , \bar{k}]$, then the last inequality and the assumption that $\{\Phi_k\}$ is bounded below imply that

$$\infty > \sum_{k=\bar{k}_2}^{\infty} (\Phi_k - \Phi_{k+1}) = \sum_{k=\bar{k}_2}^{\infty} Ared_k$$

$$\geq \sum_{k=S(\bar{k}_2,\infty)} \frac{\eta_1}{32} \epsilon_0 \min [\Delta_k , \frac{\epsilon_0}{2b_0}] .$$

This implies that

$$\liminf_{k \to \infty} \Delta_k = 0 . \tag{4.4.11}$$

This means that there exists an integer $k_3 \geq \bar{k}_2$ such that

$$\Delta_k \leq \frac{\alpha_1 \sigma_2}{a} (1 - \eta_2) \tag{4.4.12}$$

is satisfied for some $k \geq k_3$, where $a = \max [1 , \frac{32 a_0 r_*}{\epsilon_0}]$ and σ_2 is defined to be a constant that satisfies

$$\sigma_2 \;<\; \min\,[\,1\,,\; \frac{a\;\Delta_{k_3}}{\alpha_1\,(\,1-\eta_2\,)}\,,\; \frac{\epsilon_0}{2\;b_0}\,]$$

Let m be the first integer greater than k_3 such that (4.4.12) holds. This implies that $m \geq k_3+1$. So, from (3.1.3),

$$\begin{aligned} \|\,\hat{s}_{m-1}\,\|_2 \;&\leq\; \frac{\Delta_m}{\alpha_1} \\ &\leq\; \frac{\alpha_1\,\sigma_2}{\alpha_1\;a}\,(\,1-\eta_2\,) \qquad (4.4.13) \\ &\leq\; \sigma_2\,(\,1-\eta_2\,) \\ &\leq\; \sigma_2 \;<\; \frac{\epsilon_0}{2\;b_0}\,. \end{aligned}$$

But since,

$$Pred_{m-1} \;\geq\; \frac{\epsilon_0}{32}\,\min\,[\;\|\,\hat{s}_{m-1}\,\|_2\,,\; \frac{\epsilon_0}{2\;b_0}\,]\,,$$

we obtain

$$Pred_{m-1} \;\geq\; \frac{\epsilon_0}{32}\;\|\,\hat{s}_{m-1}\,\|_2\,.$$

So, by using (4.4.9), (4.4.13) and the last inequality, we get

$$\begin{aligned} |\;\frac{Ared_{m-1}-Pred_{m-1}}{Pred_{m-1}}\;| \;&\leq\; \frac{32\;a_0\;r_*\;\|\,\hat{s}_{m-1}\,\|_2}{\epsilon_0} \\ &\leq\; \frac{32\;a_0\;r_*\;\sigma_2}{\epsilon_0\;a}\,(\,1-\eta_2\,) \\ &\leq\; \sigma_2\,(\,1-\eta_2\,) \;\leq\; (\,1-\eta_2\,)\,. \end{aligned}$$

The last inequality implies that

$$\frac{Ared_{m-1}}{Pred_{m-1}} \geq \eta_2 .$$

Hence, from the rule of updating the radius of the trust region in Algorithm (3.1.2), we obtain

$$\Delta_{m-1} \leq \Delta_m .$$

This implies that $m-1$ satisfies (4.4.12) which contradicts the assumption that m is the smallest integer $\geq k_3$ such that (4.4.12) holds. Hence, for all $k \geq k_3$, we have

$$\Delta_k > \frac{\alpha_1 \sigma_2}{a} (1 - \eta_2) .$$

The last inequality contradicts (4.4.11). The supposition is wrong and hence the theorem is proven. ■

Corollary (4.22)

Under the standard assumptions. If $\{ \Phi_k \}$ is bounded below, then

$$\liminf_{k \to \infty} [\, || h_k ||_2 + || P_k \nabla l_k ||_2 \,] = 0$$

Proof

The proof follows immediately from Theorem (4.20) and Theorem (4.21). ■

From the last corollary and the termination condition in step 1 of the algorithm, we can conclude that the algorithm will terminate at a point within ϵ of a Kuhn-Tucker point.

CHAPTER FIVE

THE LOCAL ANALYSIS

In this chapter we discuss the local analysis of our algorithm when the sequence $\{x_k\}$ converges to a solution x_*. We will assume that x_* satisfies the second order sufficiency condition.

In Section 5.1 we state the local assumptions. The local analysis of our algorithm is presented in Section 5.2. It consists of three parts. In the first part of this section we study the behavior of the penalty parameter in a neighborhood of x_*. In the second part we discuss the decrease we get in the model by the trial step. The third part of this section is devoted to studying the local rate of convergence of our algorithm in a neighborhood of the minimizer x_*. We will show that, in a neighborhood of the minimizer, the algorithm will reduce to the standard SQP algorithm; hence the local rate of convergence of SQP is maintained.

5.1 THE LOCAL ASSUMPTIONS

We assume the following assumptions:

1) The sequence $\{ x_k \}$ converges to a Kuhn-Tucker point x_*.

2) x_* satisfies the second order sufficiency condition. *i.e.* there exists a λ_* such that

$$v^T \nabla_x^2 l(x_* , \lambda_*) v > 0 ,$$

for all v that satisfies $\nabla h(x_*)^T v = 0$.

3) $\nabla_x^2 l$ is Lipschitz continuous with respect to x in the neighborhood of the solution x_* .

4) There exists k_0 sufficiently large such that for all $k \geq k_0$, we have

$$|| Q_k \nabla l_k ||_2 \leq e_0 || \hat{s}_k ||_2$$

where e_0 is a constant.

Remarks

Assumption (4) is equivalent to assuming that the asymptotic progress in λ is at least of the same order as the asymptotic progress in x .

Numerical experiments have shown that for SQP, s^{QP} and $\Delta\lambda^{QP}$ have not failed to satisfy

$$|| \Delta\lambda^{QP} ||_2 \leq c || s^{QP} ||_2 , \qquad (5.1.1)$$

in the neighborhood of the solution, where " c " in this remark is used to denote a generic constant independent of k . If the step is the SQP step then inequality (5.1.1) implies Assumption (4) since $|| Q_k \nabla l_k ||_2 \leq c || \hat{\Delta\lambda}_k ||_2 + c || \hat{s}_k ||_2$. On the other hand, if $\hat{s}_k$ is the CDT step and if $|| s^{CDT} ||_2 \approx || s^{QP} ||_2$, then $|| \Delta\lambda^{CDT} ||_2$ will be near $|| \Delta\lambda^{QP} ||_2$, since $\Delta\lambda$ is linear in s , and we expect the CDT step to have the same behavior. If s^{CDT} and s^{QP} are different, we expect $\Delta\lambda^{CDT}$ to give a better progress in λ than that we get from $\Delta\lambda^{QP}$ because numerical experiments show that if s^{QP} is a bad step then $\Delta\lambda^{QP}$ will also be a bad step.

Assumption (4) and more is assumed by Gill, Murray, Saunders and Wright

(1986).

5.2 THE LOCAL ANALYSIS OF THE ALGORITHM

This section is devoted to presenting the local analysis of our algorithm when it converges to a local minimizer that satisfies the second order sufficiency condition. In Section 5.2.1 we study the behavior of the penalty parameter. We will prove that under the local assumptions the penalty parameter is bounded. In Section 5.2.2 we discuss the predicted reduction that will be obtained locally. The third part of this section is devoted to studying the local rate of convergence of our algorithm in the neighborhood of a minimizer that satisfies the second order sufficiency condition.

5.2.1) The Asymptotic Behavior of The Penalty Parameter

In this section we prove lemmas needed to study the behavior of the penalty parameter. In Lemma (5.5) we prove under the local assumptions that the penalty parameter is bounded in a neighborhood of a minimizer that satisfies the second order sufficiency condition.

Lemma (5.1)

In a neighborhood of a minimizer that satisfies the second order sufficiency condition, there exists a constant e_1 such that

$$|| P_k (\nabla l_k + B_k \hat{s}_k^q) ||_2 \geq e_1 || \hat{s}_k^p ||_2$$

where P_k , $\hat{s}_k^p$ and $\hat{s}_k^q$ are as in Corollary (4.3).

Proof

Since, using Lemma (4.8), we have

$$(\hat{s}_k^p)^T \, B_k \, \hat{s}_k^p \leq - [\, P_k \, (\, \nabla l_k + B_k \, \hat{s}_k^q \,) \,]^T \hat{s}_k^p \, . \tag{5.2.1}$$

The last inequality can be written as

$$(\hat{s}_k^p)^T \, (\, P_k \, B_k \, P_k \,) \, \hat{s}_k^p \leq - [\, P_k \, (\, \nabla l_k + B_k \, \hat{s}_k^q \,) \,]^T \hat{s}_k^p \, .$$

Now, since $(\, P_k \, B_k \, P_k \,)$ is positive definite in a neighborhood of the minimizer, then there exists a constant e_1 such that

$$e_1 \, || \, \hat{s}_k^p \, ||_2^2 \leq (\hat{s}_k^p)^T (\, P_k \, B_k \, P_k \,) \, \hat{s}_k^p \, . \tag{5.2.2}$$

So, using (5.2.1) and (5.2.2), we can write

$$e_1 \, || \, \hat{s}_k^p \, ||_2 \leq \, || \, P_k \, (\, \nabla l_k + B_k \, \hat{s}_k^q \,) \, ||_2 \, .$$

Hence we get the desired result. ■

Lemma (5.2)

In a neighborhood of a minimizer that satisfies the second order sufficiency condition, if $|| \, h_k \, ||_2 \leq e_2 \, || \, \hat{s}_k \, ||_2$ where $e_2 \leq \frac{1}{2 \, b_3}$ and b_3 is as in Lemma (4.9), then there exists a constant e_3 such that

$$|| \, P_k \, (\nabla l_k + B_k \, \hat{s}_k^q \,) \, ||_2 \geq e_3 \, || \, \hat{s}_k \, ||_2 \, .$$

Proof

Since $|| \, \hat{s}_k \, ||_2 \leq \, || \, \hat{s}_k^p \, ||_2 + \, || \, \hat{s}_k^q \, ||_2$, by using Lemma (4.9) and Lemma (5.1),

we get

$$e_1 \,||\, \hat{s}_k \,||_2 \leq \,||\, P_k\,(\nabla l_k + B_k\,\hat{s}_k^q\,)\,||_2 + e_1\, b_3\,||\, h_k\,||_2$$

$$\leq \,||\, P_k\,(\nabla l_k + B_k\,\hat{s}_k^q\,)\,||_2 + e_1\, e_2\, b_3\,||\,\hat{s}_k\,||_2\,.$$

Hence,

$$e_1\,(\,1 - e_2\, b_3\,)\,||\,\hat{s}_k\,||_2 \leq \,||\, P_k\,(\nabla l_k + B_k\,\hat{s}_k^q\,)\,||_2\,.$$

So,

$$\frac{e_1}{2}\,||\,\hat{s}_k\,||_2 \leq \,||\, P_k\,(\nabla l_k + B_k\,\hat{s}_k^q\,)\,||_2\,.$$

The result then follows if we set $e_3 = \frac{e_1}{2}$. ■

Lemma (5.3)

Let $\hat{s}_k$ be the step generated by the algorithm. Let P_k, $\overline{\Delta}_k$, $\hat{s}_k^p$ and $\hat{s}_k^q$ be as in Corollary (4.3), then for all k sufficiently large, there exists a constant e_4 such that

$$Pred_k \geq \frac{1}{4}\,||\, P_k(\nabla l_k + B_k\hat{s}_k^q\,)\,||_2 \min\,[\,\overline{\Delta}_k\,,\,\frac{||\, P_k(\nabla l_k + B_k\hat{s}_k^q\,)\,||_2}{2\,b_0}\,]$$

$$-\,e_4\,||\,\hat{s}_k\,||_2\,||\,h_k\,||_2 + r_k\,[\,||\,h_k\,||_2^2 - ||\,h_k + \nabla h_k^T\hat{s}_k\,||_2^2\,]\,. \quad (5.2.3)$$

Proof

Since, from Lemma (4.10), we have

$$Pred_k \geq \frac{1}{4}\,||\, P_k(\nabla l_k + B_k\hat{s}_k^q\,)\,||_2 \min\,[\,\overline{\Delta}_k\,,\,\frac{||\, P_k(\nabla l_k + B_k\hat{s}_k^q\,)\,||_2}{2\,b_0}\,]$$

$$-\,b_4\,||\,\hat{s}_k\,||_2\,||\,h_k\,||_2 - |\,(\nabla l_k + B_k\hat{s}_k)^T\,\overline{h}_k\,|$$

$$+ r_k \left[\| h_k \|_2^2 - \| h_k + \nabla h_k^T \hat{s}_k \|_2^2 \right].$$

Then, by using Assumption (5.1.4), for all $k \geq k_0$, we have:

$$Pred_k \geq \frac{1}{4} \| P_k(\nabla l_k + B_k \hat{s}_k^q) \|_2 \min \left[\overline{\Delta}_k , \frac{\| P_k(\nabla l_k + B_k \hat{s}_k^q) \|_2}{2 \| B \|_2} \right]$$

$$- b_4 \| \hat{s}_k \|_2 \| h_k \|_2 - (e_0 + b_0) b_8 \| \hat{s}_k \|_2 \| h_k \|_2$$

$$+ r_k \left[\| h_k \|_2^2 - \| h_k + \nabla h_k^T \hat{s}_k \|_2^2 \right].$$

Hence, if we set $e_4 = b_4 + (e_0 + b_0) b_8$, we get for all $k \geq k_0$

$$Pred_k \geq \frac{1}{4} \| P_k(\nabla l_k + B_k \hat{s}_k^q) \|_2 \min \left[\overline{\Delta}_k , \frac{\| P_k(\nabla l_k + B_k \hat{s}_k^q) \|_2}{2 b_0} \right]$$

$$- e_4 \| \hat{s}_k \|_2 \| h_k \|_2 + r_k \left[\| h_k \|_2^2 - \| h_k + \nabla h_k^T \hat{s}_k \|_2^2 \right].$$

Hence we get the desired result. ■

The first term and the third term in (5.2.3) are positive, and the second is negative. In order to prove that we will get a positive predicted reduction each iteration, we have to prove that the positive quantities are greater than or equal to the negative quantity otherwise we have to increase the penalty parameter to insure that.

Lemma (5.4)

Under the local assumptions, if $\| h_k \|_2 \leq e_5 \| \hat{s}_k \|_2$ where e_5 is chosen such that:

$$e_5 \leq \min \left[\frac{1}{2 b_3} , \frac{e_3 \min (\sqrt{3} b_0 , e_3)}{16 b_0 e_4} \right] \qquad (5.2.4)$$

where b_0 is a uniform upper bound on $|| B_k ||_2$, b_3 is as in Lemma (4.9), e_3 is as in Lemma (5.2) and e_4 is as in Lemma (5.3). Then

$$Pred_k \geq \frac{1}{8} || P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2 \min [\frac{\sqrt{3}}{2} || \hat{s}_k ||_2 , \frac{|| P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2}{2 b_0}]$$
$$+ \frac{r_k}{2} [|| h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2] . \qquad (5.2.5)$$

Proof

From Lemma (5.3), we have

$$Pred_k \geq \frac{1}{4} || P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2 \min [\overline{\Delta}_k , \frac{|| P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2}{2 b_0}]$$
$$- e_4 || \hat{s}_k ||_2 || h_k ||_2$$
$$+ r_k [|| h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2] .$$

Now,

$$\overline{\Delta}_k = \sqrt{\Delta_k^2 - || \hat{s}_k^q ||_2^2} .$$

By using Lemma (4.9) and $|| h_k ||_2 \leq \frac{1}{2 b_3} || \hat{s}_k ||_2$,

$$\overline{\Delta}_k \geq \sqrt{\Delta_k^2 - b_3^2 || h_k ||_2^2} ,$$

and we obtain

$$\overline{\Delta}_k \geq \sqrt{1 - (1/4)} || \hat{s}_k ||_2$$
$$= \frac{\sqrt{3}}{2} || \hat{s}_k ||_2 . \qquad (5.2.6)$$

Now, since $|| h_k ||_2 \leq e_5 || \hat{s}_k ||_2$ and $e_5 \leq e_2$ then by using Lemma (5.2) we have $|| P_k (\nabla l_k + B_k \hat{s}_k^q) ||_2 \geq e_3 || \hat{s}_k ||_2$, and by using (5.2.4) and (5.2.6),

we get

$$\frac{1}{8} \; || \; P_k(\nabla l_k + B_k \hat{s}_k^q) \; ||_2 \min \left[\; \overline{\Delta}_k \; , \; \frac{|| \; P_k(\nabla l_k + B_k \hat{s}_k^q) \; ||_2}{2 \; b_0} \; \right] - e_4 \; || \; \hat{s}_k \; ||_2 \; || \; h_k \; ||_2$$

$$\geq \frac{1}{8} \; e_3 \; || \; \hat{s}_k \; ||_2^2 \min \left[\; \frac{\sqrt{3}}{2} \; , \; \frac{e_3}{2 \; b_0} \; \right] - e_4 \; e_5 \; || \; \hat{s}_k \; ||_2^2$$

$$\geq 0$$

The rest of the proof follows immediately. ■

In the last lemma, we have proven that if $|| \; h_k \; ||_2 \leq e_5 \; || \; \hat{s}_k \; ||_2$, then half of the first term in (5.2.3) would cancel the second term, and the third term need never enter the calculation. This implies that if we set $r_k = r_{k-1}$, inequality (5.2.5) remains correct. So, in this case, we do not need to increase the penalty parameter.

Lemma (5.5)

Under the local assumptions, the penalty parameter is bounded.

Proof

The proof is by contradiction. Suppose that $\{ \; r_k \; \}$ is not bounded. This implies that there exists an infinite subsequence of indices $\{k_j\}$ at which $\{r_{k_j}\}$ is unbounded. Now, from Lemma (5.4), we never increase the penalty parameter if $|| \; h_k \; ||_2 \leq e_5 \; || \; \hat{s}_k \; ||_2$. So, for any $k \; \epsilon \{ \; k_j \; \}$,

$$|| \; h_k \; ||_2 > e_5 \; || \; \hat{s}_k \; ||_2 . \qquad (5.2.7)$$

Let $m \in \{ k_j \}$ and by using (4.3.1), we can write

$$\frac{r_m}{2} \frac{|| h_m ||_2}{b_1} \min [|| \hat{s}_m ||_2 , \frac{|| h_m ||_2}{b_2}]$$
$$\leq b_0 || \hat{s}_m ||_2 || \hat{s}_m^q ||_2$$
$$+ b_8 (|| Q_k \nabla l_m ||_2 + b_0 || \hat{s}_m ||_2) || h_m ||_2$$
$$+ \rho b_9 || \hat{s}_m ||_2 || h_m ||_2 .$$

If we use (5.2.7) and the local assumptions, we get

$$r_m \frac{|| \hat{s}_m ||_2}{2 b_1} \min [1 , \frac{e_5}{b_2}] \leq b_4 || \hat{s}_m ||_2$$
$$+ [b_8 (e_0 + b_0) + \rho b_9] || \hat{s}_m ||_2 ,$$

where b_4 is as in Lemma (4.10). Hence,

$$\frac{r_m}{2 b_1 b_2} \min [b_2 , e_5] \leq e_4 + \rho b_9 ,$$

where e_4 is as in Lemma (5.3). Set

$$N = [e_4 + \rho b_9] \frac{2 b_1 b_2}{\min [b_2 , e_5]} .$$

Since N is independent of m , it is an upper bound of the sequence $\{ r_{k_j} \}$ contradicting the assumption that the sequence $\{ r_{k_j} \}$ has no upper bound. This proves the theorem. ■

From the last lemma, we can conclude that for all k , $1 \leq r_k \leq r_*$ where r_* is a constant independent of k .

Since if r_k increases, it will increase by a quantity $\geq \rho$, then the number of iterations at which the penalty parameter increases is finite. Hence, there exists a constant $\bar{k}$ such that $r_k = r_{\bar{k}}$ for all $k \geq \bar{k}$.

The following theorem shows that the algorithm is well defined in the sense that at any iteration either the point (x_k , λ_k) is a Kuhn-Tucker point or the algorithm will always find an acceptable step.

Theorem (5.6)

Under the local assumptions, either the point (x_k , λ_k) is a Kuhn-Tucher point or we always find an acceptable step. *i.e.* the condition $\frac{Ared_{k+j}}{Pred_{k+j}} \geq \eta_1$ will be satisfied for some j .

Proof

If the point (x_k , λ_k) is a Kuhn-Tucker point, then there is nothing to prove. Hence, consider the case when the point (x_k , λ_k) is not a Kuhn-Tucher point. First, we assume that $|| h_k ||_2 > 0$. Since, from Corollary (4.2), we have

$$Pred_k \geq \frac{r_k}{2} \frac{|| h_k ||_2}{b_1} \min [\Delta_k , \frac{|| h_k ||_2}{b_2}] .$$

As Δ_k gets smaller, we get

$$Pred_k \geq \frac{r_k}{2} \frac{|| h_k ||_2}{b_1} \Delta_k ,$$

and since from Corollary (4.7),

$$| Ared_k - Pred_k | \leq a_0 r_k \Delta_k^2 ,$$

then, we have

$$| \frac{Ared_k - Pred_k}{Pred_k} | \leq \frac{2 a_0 b_1 \Delta_k}{|| h_k ||_2} .$$

That is,

$$\left| \frac{Ared_k}{Pred_k} - 1 \right| \leq \frac{2\, a_0\, b_1}{|| h_k ||_2} \Delta_k .$$

Now, as Δ_k again gets smaller, the quantity $\left| \frac{Ared_k}{Pred_k} - 1 \right|$ approaches 0 and hence the condition $\frac{Ared_k}{Pred_k} \geq \eta_1$ will be met after a finite number of trials.

Now, assume that $|| h_k ||_2 = 0$. Note that since we are considering the case when the point (x_k , λ_k) is feasible but not a Kuhn-Tucker point, so $|| P_k (\nabla l_k) ||_2 > 0$. From Lemma (4.10) we have

$$Pred_k \geq \frac{1}{4} || P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2 \min \left[\overline{\Delta}_k , \frac{|| P_k(\nabla l_k + B_k \hat{s}_k^q) ||_2}{2\, b_0} \right]$$

$$- b_4 || \hat{s}_k ||_2 || h_k ||_2 - | (\nabla l_k + B_k \hat{s}_k)^T \bar{h}_k |$$

$$+ r_k [|| h_k ||_2^2 - || h_k + \nabla h_k^T \hat{s}_k ||_2^2] .$$

Because $|| h_k ||_2 = 0$, $|| \hat{s}_k^q ||_2 = 0$, $\overline{\Delta}_k = \Delta_k$ and $\bar{h}_k = 0$. Thus,

$$Pred_k \geq \frac{1}{4} || P_k(\nabla l_k) ||_2 \min \left[\Delta_k , \frac{|| P_k (\nabla l_k) ||_2}{2\, b_0} \right] .$$

As Δ_k gets smaller, we get

$$Pred_k \geq \frac{1}{4} || P_k(\nabla l_k) ||_2 \Delta_k .$$

This implies, using Corollary (4.7), that

$$\left| \frac{Ared_k - Pred_k}{Pred_k} \right| \leq \frac{4\, a_0\, r_*}{|| P_k(\nabla l_k) ||_2} \Delta_k$$

So, as Δ_k gets smaller, the quantity $\left| \frac{Ared_k}{Pred_k} - 1 \right|$ approaches 0, and hence the condition $\frac{Ared_k}{Pred_k} \geq \eta_1$ will be met after a finite number of trials. This

completes the proof. ■

The last lemma implies that if at some iteration indexed k the algorithm loops infinitely without finding an acceptable step. Then the point (x_k , λ_k) is necessarily a Kuhn-Tucker point.

5.2.2) Sufficient Decrease in The Model

In this section we prove Lemma (5.7) which stated that locally the predicted reduction in the model gives at least a proportional of square of the 2-norm of the step.

Lemma (5.7)

Under the local assumptions, if $\hat{s}_k$ is the step generated by the algorithm, then, for k large enough, there exists a constant e_6 such that

$$Pred_k \geq e_6 \, || \, \hat{s}_k \, ||_2^2$$

Proof

If $|| \, h_k \, ||_2 \leq e_5 \, || \, \hat{s}_k \, ||_2$, where e_5 is as in Lemma (5.4), then from Lemma (5.4)

$$Pred_k \geq \frac{1}{8} \, || \, P_k(\nabla l_k + B_k \hat{s}_k^q) \, ||_2 \min \left[\frac{\sqrt{3}}{2} \, || \, \hat{s}_k \, ||_2 , \frac{|| \, P_k(\nabla l_k + B_k \hat{s}_k^q) \, ||_2}{2 \, b_0} \right]$$

But, since $|| \, P_k(\nabla l_k + B_k \hat{s}_k^q) \, ||_2 \geq e_3 \, || \, \hat{s}_k \, ||_2$, then

$$Pred_k \geq \frac{1}{8} e_3 \, || \, \hat{s}_k \, ||_2^2 \min \left[\frac{\sqrt{3}}{2} , \frac{e_3}{2 \, b_0} \right] .$$

On the other hand, when $|| h_k ||_2 \geq e_5 || \hat{s}_k ||_2$, we have from Corollary (4.2) that

$$Pred_k \geq \frac{1}{2} \frac{|| h_k ||_2}{b_1} \min [\Delta_k , \frac{|| h_k ||_2}{b_2}] .$$

$$\geq \frac{1}{2} \frac{e_5 || \hat{s}_k ||_2^2}{b_1} \min [1 , \frac{e_5}{b_2}] .$$

Take $e_6 = \min \{ \frac{e_3}{16 b_0} \min [\sqrt{3} b_0 , e_3] , \frac{e_5}{2 b_1 b_2} \min [b_2 , e_5] \}$, we get

$$Pred_k \geq e_6 || \hat{s}_k ||_2^2$$

Hence we get the desired result. ■

5.2.3) The Asymptotic Rate of Convergence

In this section we will assume that for each k, B_k is the exact Hessian of the Lagrangian at the point (x_k , λ_k).

We start this section by proving Theorem (5.8) which is needed to study the local rate of convergence. In Theorem (5.9) we prove under the local assumptions, for k sufficiently large, the SQP steps will always be taken. So, the strategy of taking s^{QP}, if possible, will make our algorithm, for large k, produce the SQP steps. Hence, for large k, the steps are the SQP steps and consequently the convergence rate of (x_k , λ_k) to (x_* , λ_*) is q-quadratic.

Theorem (5.8)

Under the local assumptions, if $\hat{s}_k$ is the steps generated by the algorithm, then

there exists k_1 such that for all $k \geq k_1$, we have

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

That is, for all k large enough, the trust region radius, Δ_k, will be inactive.

Proof

Since, for some $\xi \in (0, 1)$,

$$\begin{aligned} L(x_k+\hat{s}_k, \lambda_k+\hat{\Delta\lambda}_k; r_k) &= L(x_k, \lambda_k+\hat{\Delta\lambda}_k; r_k) + \nabla_x L(x_k, \lambda_k+\hat{\Delta\lambda}_k; r_k)^T \hat{s}_k \\ &\quad + \frac{1}{2} \hat{s}_k^T \nabla_x^2 L(x_k + \xi \hat{s}_k, \lambda_k+\hat{\Delta\lambda}_k, r_k)^T \hat{s}_k \end{aligned}$$

$$\begin{aligned} &= L(x_k, \lambda_k; r_k) + \hat{\Delta\lambda}_k^T(h_k + \nabla h_k^T \hat{s}_k) \\ &+ \nabla_x l(x_k, \lambda_k) + \frac{1}{2} \hat{s}_k^T B_k \hat{s}_k \\ &+ r_k [\, ||h_k + \nabla h_k^T \hat{s}_k||_2^2 - ||h_k||_2^2 \,] \\ &+ \frac{1}{2} \hat{s}_k^T [\, \nabla_x^2 L(x_k + \xi\hat{s}_k, \lambda_k+\hat{\Delta\lambda}_k; r_k) - \nabla_x^2 L(x_k, \lambda_k+\hat{\Delta\lambda}_k; r_k) \,]^T \hat{s}_k \\ &+ \frac{1}{2} \hat{s}_k^T \nabla^2 h_k \hat{\Delta\lambda}_k \hat{s}_k + r_k \hat{s}_k^T \nabla^2 h_k h_k \hat{s}_k . \end{aligned}$$

Hence,

$$\begin{aligned} Ared_k &\geq -\nabla l_k^T \hat{s}_k - \frac{1}{2} \hat{s}_k^T B_k \hat{s}_k - \hat{\Delta\lambda}_k^T(h_k + \nabla h_k^T \hat{s}_k) \\ &\quad + r_k [\, ||h_k||_2^2 - ||h_k + \nabla h_k^T \hat{s}_k||_2^2 \,] \\ &\quad - o(||\hat{s}_k||_2^2) - \frac{1}{2} |\hat{s}_k^T \nabla^2 h_k \hat{\Delta\lambda}_k \hat{s}_k| - r_* |\hat{s}_k^T \nabla^2 h_k h_k \hat{s}_k| . \end{aligned}$$

Using Lemma (5.7), for k large enough, we have

$$\frac{Ared_k}{Pred_k} \geq 1 - \frac{1}{e_6} \left[\frac{o(||\hat{s}_k||_2^2)}{||\hat{s}_k||_2^2} + \frac{|\hat{s}_k^T \nabla^2 h_k \hat{\Delta\lambda}_k \hat{s}_k|}{||\hat{s}_k||_2^2} + \frac{r_* |\hat{s}_k^T \nabla^2 h_k h_k \hat{s}_k|}{||\hat{s}_k||_2^2} \right] .$$

But, since by the local assumptions $||s_k||_2 \to 0$, $||\nabla l_k||_2 \to 0$, and

$|| h_k ||_2 \rightarrow 0$, then the quantities $\frac{o (|| \hat{s}_k ||_2^2)}{|| \hat{s}_k ||_2^2}$, $\frac{| \hat{s}_k^T \nabla^2 h_k \, \hat{\Delta\lambda}_k \, \hat{s}_k |}{|| \hat{s}_k ||_2^2}$, and $\frac{r_* \, | \hat{s}_k^T \nabla^2 h_k \, h_k \, \hat{s}_k |}{|| \hat{s}_k ||_2^2}$ are arbitrary small for k sufficiently large. Hence, there exists an integer k_1 such that for all $k \geq k_1$, we have

$$1 - [\frac{o (|| \hat{s}_k ||_2^2)}{|| \hat{s}_k ||_2^2} + \frac{| \hat{s}_k^T \nabla^2 h_k \, \hat{\Delta\lambda}_k \, \hat{s}_k |}{|| \hat{s}_k ||_2^2} + \frac{\rho \, | \hat{s}_k^T \nabla^2 h_k \, h_k \, \hat{s}_k |}{|| \hat{s}_k ||_2^2}] > \eta_2 .$$

Consequently, for all $k \geq k_1$, we have

$$\frac{Ared_k}{Pred_k} \geq \eta_2 . \tag{5.3.1}$$

The last inequality implies that the trust region radius Δ_k for $k \geq k_1$ is updated according to the rule

$$\Delta_{k+1} = \max [\Delta_k , \alpha_3 || s_k ||_2] .$$

Hence, $\Delta_k \geq \Delta_{k_1}$ for all $k \geq k_1$ and using the assumption that $|| s_k ||_2 \rightarrow 0$ we can conclude that there exists an integer $k_2 \geq k_1$ such that the trust region is inactive for all $k \geq k_2$. Hence we get the desired result.

■

Theorem (5.9)

Under the local assumptions, for k sufficiently large, the SQP steps will be taken and consequently (x_k , λ_k) converges to (x_* , λ_*) q-quadratically.

Proof

From the last lemma, $\Delta_k \geq \Delta_{k_1}$ for all $k \geq k_1$. Now suppose there exists an integer $k_3 \geq k_1$ such that $s_k \neq s_k^{QP}$ for all $k \geq k_3$. This implies that, for all

$k \geq k_3$

$$|| s_k^{QP} ||_2 > \Delta_k \geq \Delta_{k_3}$$

which contradict the fact that $|| s_k^{QP} ||_2 \rightarrow 0$. Therefore, there exists at least one step $s_{k_j} = s_{k_j}^{QP}$ where $k_j \geq k_3$.

Let k_4 be the smallest integer greater than k_3 such that $s_{k_4} = s_{k_4}^{QP}$, and such that SQP method generates steps that satisfies

$$|| u_{k+1} ||_2 \leq m_1 || u_k ||_2^2 ,$$

where $u_k = \begin{bmatrix} s_k^{QP} \\ \Delta\lambda_k^{QP} \end{bmatrix}$ and m_1 is a constant.

But, since the SQP steps $\{ s_k^{QP} \}$ converge r-quadratically. This implies that, for all $k \geq k_4$, we have

$$|| s_k^{QP} ||_2 \leq m_2 (\alpha^2)^{k_4} .$$

where m_2, α are constants and $\alpha < 1$. This means that if we choose k_4 sufficiently large such that

$$m_2 (\alpha^2)^{k_4} \leq \Delta_{k_4} .$$

Then, $|| s_{k_4}^{QP} ||_2 \leq \Delta_{k_4}$ and for all $k \geq k_4$, we have

$$|| s_k^{QP} ||_2 \leq \Delta_{k_4} .$$

But since, for $k \geq k_1$, we have $\Delta_k \leq \Delta_{k+1}$, then

$$|| s_{k_4+1}^{QP} ||_2 \leq \Delta_{k_4} \leq \Delta_{k_4+1} .$$

The last inequality and the fact that for all $k \geq k_2$ all the steps are acceptable steps imply that

$$s_{k_4+1} = s_{k_4+1}^{QP} .$$

By induction, for all $k \geq k_4$, we can conclude that

$$s_k = s_k^{QP} .$$

This means that the sequence $\{ x_k , k \geq k_4 \}$ generated by the algorithm is the sequence of the SQP iterates and consequently the local rate of convergence is q-quadratic. ■

CHAPTER SIX

CONCLUDING REMARKS

We have considered a trust region algorithm for solving the equality constrained optimization problem. This algorithm is a variant of the 1984 Celis-Dennis-Tapia algorithm. We have presented a global and local convergence analysis for this algorithm.

Our global convergence theory is sufficiently general that it holds for any algorithm that generates steps that give at least a fraction of Cauchy decrease in the quadratic model of the constraints.

The subproblem that has to be solved at each iteration is not in general the successive quadratic programming subproblem. However, we have shown that under mild assumptions, in the neighborhood of the minimizer, the algorithm will reduce to the standard SQP algorithm; hence the local rate of convergence of the SQP in maintained.

The augmented Lagrangian function was used as a merit function. A scheme for updating the penalty parameter was presented. The behavior of the penalty parameter was discussed.

For future work, there are many questions that should be answered:

Although intensive numerical experiences with the CDT algorithm were reported by Celis, Dennis and Tapia (1984), Celis (1985) and Celis, Dennis and

Tapia (1987), we believe that the implementation of the algorithm must be refined. In particular, an efficient algorithm for solving the CDT subproblem is needed. This will require a closer look at the CDT subproblem and the characteristics of its solution. Currently, this is the topic of much research, *e.g.* Yuan (1987), but the problem has not been solved.

A related important question that has to be looked at is how to approximate the Hessian of the Lagrangian in order to be used to produce an efficient algorithm.

Another important research topic that should be considered is how to generalize this approach to handle the inclusion of nonlinear inequality constraints in the problem.

REFERENCES

Armijo, L.[[1966]. Minimization of Functions Having Lipschitz-Continuous Friest Partial Derivatives; Pacific J. Math., 16, pp.1-3.

Bartholomew-Biggs, M.C.[1980]. Recursive Quadratic Programming Based on Penalty Function for Constrained Optimization; Nonlinear Optimization, Theory and Algorithms, L.C.W. Dixon, E. Spedicato, G.P. Szego, eds., Birkhauser.

Bartholomew-Biggs, M.C.[1982]. Recursive Quadratic Programming Methods for Nonlinear Constraints; Nonlinear Optimization 1981, M.J.D.Powel,ed., Academic Press London.

Bertsekas, D.P.[1976]. On Penalty and Multiplier Methods for Constrained Optimization; SIAM J. Control and Optimization, 14, pp.216-135

Bertsekas, D.P.[1982]. Constrained Optimization and Lagrange Multiplier Methods; Academic Press, New York, NY.

Boggs, P.T. and Tolle, J.W.[1980]. Augmented Lagrangians that are Quadratic in the Multiplier; JOTA, 3, pp.17-26.

Boggs, P.T. and Tolle, J.W.[1984]. A Family of Descent Functions for Constrained Optimization; SIAM J. Numerical Analysis, 21, pp.1146-1161

Buys, J.D.[1972]. Dual Algorithms for Constrained Optimization Problems; Ph.D. Thesis, University of Leiden, Netherlands.

Byrd, R.H.[1984]. On the Convergence of Constrained Optimization Methods with Accurate Hessian Information on a Subspace; Dept. of Computer Sc. C.U. Boulder Technical Report # CU-CS-270-84.

Byrd, R.H.[1985]. An Example of Irregular Convergence in Some Constrained Optimization Methods That Use Projected Hessian; Math. Prog., 32, pp.232-237.

Byrd, R.H., Schnabel, R.B. and Shultz, G.A.[1985]. A Trust Region Algorithm for Nonlinearly Constrained Optimization; Dept. of Computer Science, Univ.of Colorado at Boulder Report No.CU-CS-313-85,Bouder Colorado

Byrd, R.H., Omojokun, E.O., Schnabel, R.B. and Shultz, G.A.[1987]. Robust Trust Region Methods for Nonlinearly Constrained Optimization, Present at the 1987 SIAM Conference on Optimization, Houston, TX (to appear).

Burke, J.V. and Han, S-P.[1985]. A Modified Sequential Quadratic Programming Method. (to appear).

Celis, M.R.[1985]. A Trust Region Strategy for Nonlinear Equality Constrained Optimization; Ph.D. thesis, Rice University.

Celis, M.R, Dennis, J.E. and Tapia, R.A.[1984]. A Trust Region Strategy for Nonlinear Equality Constrained Optimization; in Numer.Opt.(Boggs,Byrd and Schnabel,ed), SAIM Philadelphia, pp.71-82.

Celis, M.R, Dennis, J.E., and Tapia, R.A.[1987]. An Algorithm Based on a Convenient Trust-Region Subproblem for Nonlinear Programming; Presented at the 1987 SIAM 35th. Anniversary Meeting, 1987, Denver, CO. (to appear)

Chamberlain, R.M., Lemarechal, C., Pedersen, H.C. and Powell, M.J.D.[1982]. The Watchdog Technique for Forcing Convergence in Algorithms for Constrained Optimization; Mathematical Programming study, 16, pp.1-17.

Coleman,T.R. and Conn, A.R[1982]. Nonlinear Programming Via an Exact Penalty Function: Asymptotic analysis; Math. Prog., 24, pp.123-136.

Coleman, T.F. and Conn, A.R.[1984]. On the Local Convergence of a Quasi-Newton Method for the Nonlinear Programming Problems; SIAM J. Numerical Analysis, 21, pp.755-769.

Conn, A.R., Gould, N.I.M. and Toint, P.L.[1986a]. Global Convergence of a Class of Trust Region Algorithms for Optimization With Simple Bounds; University of Waterloo Computer Science Dept.Report No 86/1, Waterloo, Canada.

Conn, A.R., Gould, N.I.M., and Toint, P.L.[1986b]. Testing a Class of Methods for Solving Minimization Problems With Simple Bounds on the Variables; Univ. of Waterloo Computer Science Dept.Report No 86/3, Waterloo, Canada.

Courant, R.[1943]. Variational Methods for the Solution of Problems of Equilibrium and Vibrations; Bull.Amer.Math.Soc, 49, pp.1-23.

Dennis, J.E. and Moré, J.J.[1977]. Quasi-Newton Methods: Motivation and theory; SIAM Review, 10, pp.46-89.

Dennis, J.E. and Schnabel, R.B.[1983]. Numerical Methods for Unconstrained Optimization and Nonlinear Equations; Prentice-Hall, Englewood Cliffs,N.J.

Dennis, J.E., El-Alem, M.M. and Tapia, R.A.[1987]. Global Convergence Analysis For The Celis-Dennis-Tapia Algorithm; Presented at the SIAM 35th. Anniversary Meeting, 1987, Denver, CO. (to appear)

Di Pillo, G. and Grippo, L.[1979]. A New Class of Augmented Lagrangians in Nonlinear Programming; SIAM J. Control Optimization, 17, pp.618-628.

Fiacco, A.V., and McCormick, G.P.[1968]. Nonlinear Programming: Sequential Unconstrained Minimization Techniques; John Wiley and Sons, New York.

Fletcher, R.[1973]. An Exact Penalty Function for Nonlinear Programming with Inequalities; Math. Programming, 5, pp.129-150.

Fletcher, R.[1981]. Practical Methods of Optimization , vol 2, Constrained Optimization ; Wiley, Chichester, 1981.

Fletcher, R.[1982]. Second Order Correction for Non-differentiable Optimization. Numerical Analysis; Lecture Notes in Mathematics, 912. Watson, G.A, ed., Springer-Verlag.

Fletcher, R.[1984]. An L1-Penalty Method for Nonlinear Constraints; in Numerical Optimization (Boggs,Byrd and Schnabel,ed), SIAM Philadelphia, pp.26-40

Garcia-Palomares, U.M. and Mangasarian, O.L.[1976]. Superlinearly Convergent Quasi-Newton Algorithms for Nonlinearly Constrained Optimization Problems; Math. Prog., 11, pp.1-13.

Gay D.M.[1983]. A Trust Region Approach to Linearly Constrained Optimization, (to appear)

Gilbert, J.C.[1986]. On the Local and Global Convergence of a Reduced Quasi-Newton Method; TR. Aug. 1987, INRIA, France.

Gill, P.E., Murray, W., and Wright, M.H.[1981]. Practical Optimization; Academic Press, London, New York.

Gill, P.E., Murray, W., Saunders, M.A, and Wright, M.[1986]. Some Theoretical Properties of an Augmented Lagrangian Merit Function; Report SOL 86-6. Stanford University.

Goldstein, A.A.[1967]. Constructive Real Analysis, Harper & Row, New York, NY.

Goodman, J.[1985]. Newton's Method for Constrained Optimization; Math. Programming, 33, PP 162-171.

Gurwitz, C.B.[1986]. Sequential Quadratic Programming Methods Based on Approximating a Projected Hessian Matrix ; Computer Sc. Dept. Tech. Report# 219. NYU, New York.

Haarhoff, P.C. and Buys, J.D.[1970]. A new method for the optimization of a non-linear function subject to nonlinear constraints; Computer Journal, 13, pp.178-184.

Han, S.P.[1976]. Superlinear Convergent Variable Metric Algorithms for general nonlinear programming problems; Math.Prog., 11, pp.263-283.

Han, S.P.[1977a]. Dual Variable Metric Algorithms for Constrained Optimization; SIAM J. Control and Opt., 15, pp546-565.

Han, S.P.[1977b]. A Globally Convergent Method for Nonlinear Programming; JOTA, 22, pp.297-309.

Hestenes, M.R.[1969]. Multiplier and Gradient Methods; JOTA, 4, pp.303-320.

Levenberg, K.[1944]. A Method for the Solution of Certain Problems in Least-Squares; Quaterly J. of Applied Math., l2, pp.164-168.

Maratos, N.[1978]. Exact Penalty Functions for Finite Dimensional and Control Optimization Problems. Ph.D. Thesis, University of London

Marquardt, D.[1963]. An Algorithm for Least-Squares Estimation of Nonlinear Parameters; SIAM J. on Applied Math., 11, pp.431-441.

Mayne, D.Q. and Polak, E.[1982]. A Superlinearly Convergent Algorithm for Constrained Optimization Problems; Math. Programming Stud., 16, pp.45-61.

Miele, A, Cragg, E.E., Iyer, R.R., and Levy, A.V.[1971a]. Use of the Augmented Penalty Function in Mathematical Programming, part I; J. Opt. Th. Appl., 8, pp.115-130.

Miele, A, Cragg, E.E., Iyer, R.R., and Levy, A.V.[1971b]. Use of the Augmented Penalty Function in Mathematical Programming, part II; J. Opt. Th. Appl., 8, pp.131-153.

Miele, A., Moseley, P.E., and Cragg, E.E.[1972a]. A Modification of the Method of Multipliers for Mathematical Programming; Techniques of Optimization, A. Balakrishnan,ed., Academic Press, New York, NY.

Miele, A., Moseley, P.E., Levy, A.V., and Coggins, G.M.[1972b]. On the Method of Multipliers for Mathematical Programming Problems; J. Optimization Theory and Applications, 10, pp.1-33.

Moré, J.J.[1977]. The Levenberg-Marquardt Algorithm; Implementation and Theory; in G.A.Watson,ed.,Numerical Analysis, Lecture None in Mathematics 630, Springer-Verlag, Berlin.

Moré, J.J. and Sorensen, D.C.[1983]. Computing A Trust Region Step; SIAM J. Sci. Stat. Comput., l4, No3, pp.553-572.

Moré, J.J.[1982]. Recent Developments in Algorithms and Software for Trust Region Methods; Mathematical Programming-The state of the Art, Born 1982 (Bachem,A, Grotschel, and Korte, B. ed.), Springer-Verlag. New York. pp.258-287

Murray, W. and Wright, M.H.[1982]. Computation of the Search Director in Constrained Optimization Algorithms; Math.Prog. Study, 16, pp.62-83.

Powell, M.J.D[1977]. A Fast Algorithm for Nonlinearly Constrained Optimization Calculation; Report DAMTP 77/NA 2, Cambridge University, England.

Powell, M.J.D [1978]. The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculation; in Nonlinear Programming 3, (Mangasarian, O., Meyer, R., and Robinsons, S. eds.), pp.27-63, Academic Press, New York.

Powell, M.J.D.[1986]. A method for Nonlinear Constraints in Minimization Calculations; Presented at the 1986 IMA/SIAM Meeting on "The State of the Art

in Numerical Analysis".

Powell, M.J.D[1984]. A Recursive Quadratic Programming Algorithm that Uses Differentiable Exact Penalty Functions; Math. Programming (to be published).

Powell, M.J.D and Yuan, Y.[1986]. A Trust Region Algorithm for Equality Constrained Optimization; Report DAMTP 1986/NA2, Cambridge, England.

Schittkowski, K.[1983]. On the Convergence of a Sequential Quadratic Programming Method with an Augmented Lagrangian Line Search Function; Math. Operationsforschung U. Statistik, Ser. Optimization, 14, pp.197-216.

Shultz, G.A.[1983]. Computationally Practical Globally Convergent Algorithms for Unconstrained and Linearly Constrained Optimization; Ph.D.Thesis, Dept.of Maths, Univ.of Colorado at Boulder.

Shultz, G.A, Schnabel, R.B., and Byrd, R.H.[1985]. A Family of Trust-Region-Based Algorithms for Unconstrained Minimization with Strong Global Convergence Properties; SIAM J.Numer.Anal., 22, No1, pp.47-67.

Steihaug, T.[1983]. The Conjugate Gradient Method and Trust Regions in Large Scale Optimization; SIAM J.Numer.Anal, 20, No3, pp.626-637.

Sorensen, D.C.[1982]. Newton's Method with a Model Trust Region Modification; SIAM J. Numer. Anal., 19, No2, pp.409-426.

Tapia, R.A.[1974a]. Newton's Method for Problems with Equality Constraints; SIAM J. Numerical Analysis, 11, pp.174-196.

Tapia, R.A.[1974b]. Newton's Method for Optimization Problems with Equality Constraints. SIAM J. Numerical Analysis, 11, pp.874-886.

Tapia, R.A.[1977]. Diagonalized Multiplier Methods and Quasi-Newton Methods for Constrained Optimization; J.of Opt.Th.and Applications, 22, pp.135-194.

Tapia, R.A.[1978]. Quasi-Newton Methods for Equality Constrained Optimization: Equivalence of Existing Methods and a New Implementation; in Nonlinear Programming 3, (Mang, O., Meyer, R., and Rob, S. eds.) pp.125-164, Academic Press, New York.

Tapia, R.A.[1983]. An Introduction to the Algorithms and Theory of Constrained Optimization; Unpublished notes, Rice University, Houston, Tx.

Tolle, J.W and Boggs, P.T.[1987]. Merit Functions and Nonlinear Programming, Presented at the 1987 SIAM Conference on Optimization, Houston, TX. (to appear)

Vardi, A.[1985]. A Trust Region Algorithm for Equality Constrained Minimization: Convergence Properties and Implementation; SIAM J.Numer.Anal., 22, No3, pp.575-591

Wolfe, P.[1969]. Convergence conditions for ascent Methods; SIAM Review, 11, pp.226-235.

Wolfe, P.[1969]. Convergence Conditions for Ascent Methods II: Some Corrections; SIAM Review, 13, pp.185-188.

Yuan, Y.[1987]. On a Subproblem of Trust Region Algorithms For Constrained Optimization; Report DAMTP 1987/NA10, Cambridge, England.