INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
A global convergence theory for a general class of trust region algorithms for equality constrained optimization

Maciel, María Cristina, Ph.D.

Rice University, 1993
PLEASE NOTE

The original document received by U.M.I. contained poor print. Pages were filmed as received.

This reproduction is the best copy available.

University Microfilms International
RICE UNIVERSITY

A Global Convergence Theory for a General Class of Trust Region Algorithms for Equality Constrained Optimization

by

María Cristina Maciel

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE
Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

J.E. Dennis, Jr., Chairman
Noah Harding Professor of Mathematical Sciences

Richard A. Tapia.
Noah Harding Professor of Mathematical Sciences

Mark R. Wiesner
Assistant Professor of Environmental Science and Engineering

Robert Michael Lewis
Research Scientist

Mahmoud El-Alem
Assistant Professor of Mathematics,
Alexandria University, Egypt

Houston, Texas
August, 1992
A Global Convergence Theory for a General Class of Trust Region Algorithms for Equality Constrained Optimization

María Cristina Maciel

Abstract

This work is concerned with global convergence results for a broad class of trust region sequential quadratic programming algorithms for the smooth nonlinear programming problem with equality constraints. The family of algorithms to which our results apply is characterized by very mild conditions on the normal and tangential components of the steps that its members generate. The normal component must predict a fraction of Cauchy decrease condition on the quadratic model of the linearized constraints. The tangential component must predict a fraction of Cauchy decrease condition on the quadratic model of the Lagrangian function associated with the problem in the tangent space of the constraints. The other main characteristic of this class of algorithms is that the trial step is evaluated for acceptance by using as merit function the Fletcher exact penalty function with a penalty parameter specified by El-Alem. The properties of the step together with the way that the penalty parameter is chosen allow us to establish that while the algorithm does not terminate, the sequence of trust region radii is bounded away from zero and the nondecreasing sequence of penalty parameters is eventually constant. These results lead us to conclude that the algorithms are well defined and that they are globally convergent.

The class includes well-known algorithms based on the Celis-Dennis-Tapia subproblem and on the Vardi subproblem. As an example we present an algorithm which can be viewed as a generalization of the Steihaug-Toint dogleg method for the unconstrained case. It is based on a quadratic programming algorithm that uses as feasible point a step in the normal direction to the tangent space of the constraints and then does feasible conjugate reduced-gradient steps to solve the quadratic program. This algorithm should cope quite well with large problems.
Acknowledgments

First, I would like to thank my family; without their emotional support this degree would not have been possible.

I am grateful to the members of my thesis committee for their time and suggestions. I wish to express my gratitude to my advisor, professor John E. Dennis Jr., for his generous support, guidance and patience. I would also like to thank Dr. Mahmoud El-Alem for his suggestions which help to improve the final version of this work and Dr. Michael R. Lewis for his careful reading of the first draft. I am very grateful to professor Richard Tapia not only for his assistance and advise but also for his friendship during all the years that I have spent in Texas.

I owe a great deal of thanks to my friends from Argentina. Very special thanks go to Leticia Giretti, Olga Rueda, Vilma Viazzi and Alicia de Weerth for their encouragement and support when I first came to this country.

Thanks go to Martha Contreras, Debora Cores, Amr El-Bakry, Rosemary Garcia, Marcos Raydan and Marcela Rosenblum who shared with me most of my graduate years in Rice. I am specially indebted with Amr for listening patiently to almost all the ideas of this work and for tolerating me as his officemate.

Finally, I would like to thank Francesco Zirilli for his affection, understanding and especial sense of humor. At the times when I questioned my will to get my degree, his advise made me smile and keep on working.
A mi padres
Olga E. Baez y Alberto D. Maciel,

a mi hermano
Oscar A. Maciel.
Contents

Abstract ii
Acknowledgments iii
List of Illustrations vii

1 Introduction 1
1.1 Overview of the work .................. 3

2 Trust region methods for unconstrained optimization 4
2.1 The general trust region algorithm for (UCMIN) ............ 5
2.2 Global convergence results .................. 7
2.3 The Steihaug-Toint algorithm for unconstrained minimization 10

3 Trust region methods for constrained optimization 13
3.1 The general trust region algorithm for (EQC) ............... 13
3.2 The successive quadratic programming method .......... 14
3.2.1 The quadratic model $q_e(s)$ ............... 14
3.3 Existing trust region subproblems for (EQC) ............. 15
3.3.1 The tangent space approach ............... 16
3.3.2 The full space approach ............... 19
3.4 The general trial step .................. 22
3.5 The choice of the merit function .................. 25
3.6 Evaluating the trial step .................. 25
3.7 The penalty parameter .................. 28
3.7.1 Some existing schemes for updating the penalty parameter 29
3.7.2 The choice of the penalty parameter ............... 30
3.8 Termination of the algorithm ............... 32
3.9 Statement of the algorithm ............... 32
4 A trust region algorithm based on the tangent space approach

4.1 A conjugate direction algorithm for equality constrained quadratic programming ........................................... 35
   4.1.1 The conjugate reduced-gradient algorithm .................. 36
   4.1.2 Properties of the reducer operator ......................... 39

4.2 Choices for $\Theta_c$ .................................................. 40
   4.2.1 Via conjugate reduced gradient algorithm ................. 42
   4.2.2 Via Craig's method ........................................... 42

4.3 Computation of a trial step ....................................... 43

4.4 Statement of the algorithm ....................................... 45

5 Towards the global convergence theory .......................... 47

5.1 The standard assumptions ........................................ 47
   5.1.1 Properties .................................................. 48

5.2 Properties of the trial step ....................................... 52

5.3 The decrease in the model ........................................ 59

6 The behavior of the penalty parameter ......................... 68

7 The global convergence theorems ................................ 77

7.1 The finite termination theorem ................................ 77

7.2 Properties of the sequence $\delta_k$ ............................ 78

7.3 The global convergence theorem ................................ 84

8 Concluding remarks ................................................ 88

8.1 Current and future work ....................................... 88

A An algorithm for computing a trial step ....................... 90

Bibliography ................................................................ 93
Illustrations

3.1 Vardi subproblem. .................................................. 18
3.2 Powell-Yuan subproblem. ........................................ 19
3.3 Celis-Dennis-Tapia subproblem. ............................... 20
3.4 The general trial step. ............................................ 23
3.5 Fraction of Cauchy decrease on the $\mathcal{N}(\nabla C^T_c)$. .................................................. 24

4.1 $\|s^m_c\| \leq r\delta_c$. ............................................ 40
4.2 $\|s^m_c\| \geq \delta_c$. ............................................ 41
4.3 $r\delta_c < \|s^m_c\| < \delta_c$. .................................... 41
Chapter 1

Introduction

This work is concerned with the development of a global convergence theory for a broad class of algorithms for the equality constrained minimization problem

\[
\begin{align*}
\text{(EQC)} & \equiv \begin{cases} 
\text{minimize} & f(x) \\
\text{subject to} & C(x) = 0,
\end{cases}
\end{align*}
\]

where \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) and \( C : \mathbb{R}^n \rightarrow \mathbb{R}^m \) such that \( C(x) = (c_1(x), \ldots, c_m(x))^T \) are at least twice continuously differentiable, and \( m \leq n \).

Let us begin by introducing some basic definitions and notation which will be used through this work.

**Definition 1.1** (The feasible set)
The set \( \mathcal{F} = \{ x \in \mathbb{R}^n : C(x) = 0 \} \) is said to be the feasible set for problem (EQC).

**Definition 1.2** (Lagrangian function)
The function \( \ell : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R} \) defined as \( \ell(x, \lambda) = f(x) + \lambda^T C(x) \) is said to be the Lagrangian function associated with problem (EQC).
The vector \( \lambda \in \mathbb{R}^m \) is said to be the vector of Lagrange multipliers and its component \( \lambda_i \); the Lagrange multiplier associated with the constraint \( c_i(x) \).

In order to establish necessary conditions in terms of the Lagrangian function a constraint qualification is required.

**Definition 1.3** (Regularity)
A feasible point \( x_* \) for problem (EQC) is said to be regular if the set of gradients \( \{ \nabla c_1(x_*), \ldots, \nabla c_m(x_*) \} \) is linearly independent.

**First Order Necessary Conditions** for a regular point \( x_* \) to be a solution of (EQC) are that there exists a Lagrange multiplier \( \lambda_* \) such that \( (x_*, \lambda_*) \) satisfies:
(a) $\nabla_x \ell(x_*, \lambda_*) = 0$

(b) $C(x_*) = 0$.

The conditions (a) and (b) are said to be the Kuhn-Tucker Conditions for problem (EQC).

The Second Order Necessary Conditions for a regular point $x_*$ to be a solution of (EQC) are

a) The first order necessary conditions hold at $x_*$.

b) The Hessian matrix of the Lagrangian function is positive semidefinite on the tangent space of the constraints, that is

$$z^T \nabla_x^2 \ell(x_*, \lambda_*) z \geq 0 \quad \text{whenever } z \in \mathcal{N}(\nabla C^T(x_*))$$

where

$$\mathcal{N}(\nabla C^T(x_*)) = \{ z : z \in \mathbb{R}^n, \nabla C^T(x_*) z = 0 \}.$$ 

The global convergence theory that we establish in this work holds for a class of nonlinear programming algorithms for (EQC') which is characterized by the following features:

1. The algorithms of the family use the trust region approach as a globalization strategy.

2. All these algorithms generate steps that satisfy very mild conditions on the trial steps' normal and tangential components. The normal component satisfies a fraction of Cauchy decrease condition on the quadratic model of the linearized constraints. The tangential component satisfies a fraction of Cauchy decrease on the quadratic model of the Lagrangian function associated with (EQC).

3. The other main characteristic of this class of algorithms is that the step is evaluated for acceptance by using the Fletcher exact penalty function and the penalty parameter is updated by using the scheme proposed by El-Alem in 1991 [9].

Concepts like trust region strategy, fraction of Cauchy decrease and merit function are the key points in this research and we will deal with them in the next chapters. This theory is developed for the equality constrained case, but it can be applied to the general case. The algorithms of this family are based on the well-known successive
quadratic programming approach, which copes with inequality constraints by one of the strategies known as EQP and IQP. In the first strategy the choice of the of the current active set is made outside the procedure that determines the step, so in this case the trial step is found by solving an equality constrained quadratic programming problem. Algorithms based on the IQP strategy choose the active set inside the procedure that determine the step, then the trial step solves a quadratic programming problem with inequality constraints. We believe that if the problem is large the choice should be EQP, because the large scale inequality constrained quadratic programming problem is very expensive compared to the equality constrained case.

In the remainder of this work the following notation is used: the sequence of points generated by an algorithm is denoted by \( \{x_k\} \). This work also uses subscripts -, c and + to denote the previous, the current and the next iterates. However, when we need to work with a whole sequence we will use the index \( k \). The matrix \( H(x, \lambda) \) denotes an approximation of the Hessian of the Lagrangian function with respect to \( x \), that will be defined later in Chapter 5. Subscripted values of functions mean the function evaluated at a particular point; for example, \( \ell_c = \ell(x_c, \lambda_c) \). Finally all the norms will be \( \ell_2 \)-norms.

1.1 Overview of the work

In Chapter 2 we present the basic ideas of trust region method for the unconstrained minimization problem. We will introduce the concept of fraction of Cauchy decrease condition and discuss its importance in the global convergence theory. Chapter 3 is devoted to presenting all the elements for the nonlinear programming algorithm based on the trust region approach. Also we will survey some existing trust region subproblems found in the literature and discuss our choice of the merit function, the well-known exact penalty function proposed by Fletcher in 1972 [11]. The fourth chapter is devoted to developing an algorithm for finding a trial step. It can be view as a generalization to (EQC) of the Steihaug-Toint dogleg method for the unconstrained case. This algorithm should work quite well for large problems. In Chapter 5 we will state the assumptions under which the global convergence theory is established. Chapter 6 is devoted to a discussion of the behavior of the penalty parameter, and in Chapter 7 we present the global convergence results. Finally we will make some concluding remarks in Chapter 8.
Chapter 2

Trust region methods for unconstrained optimization

Trust region algorithms have proved to be very successful for solving the smooth unconstrained minimization problems

\[
\text{(UMIN)} \equiv \begin{cases} 
\text{minimize} & f(x) \\
\text{subject to} & x \in \mathbb{R}^n, 
\end{cases}
\]

where \( f : \mathbb{R}^n \to \mathbb{R} \) is a continuously differentiable function. These algorithms are practically reliable and theoretically satisfactory. For a complete survey see Moré [16] and the book of Dennis and Schnabel [5].

Briefly the idea of the trust region strategy for the unconstrained case is as follows: If \( x_c \) is the current iterate, the function \( f \) is approximated in a neighborhood of \( x_c \) by a quadratic model function

\[
q_c(s) = f_c + \nabla f_c^T s + \frac{1}{2} s^T G_c s
\]

\[
\approx f(x_c + s),
\]

where \( G_c \) is the Hessian matrix \( \nabla^2 f(x_c) \) or an approximation to it.

The trust region step, the trial step, is obtained as an approximate solution to the trust region subproblem:

\[
\text{(TRS)} \equiv \begin{cases} 
\text{minimize} & q_c(s) \\
\text{subject to} & ||s||_2 \leq \delta_c, 
\end{cases}
\]

where \( \delta_c > 0 \) is the given trust region radius. As we said, in this work we are using the \( \ell_2 \)-norm, however in many cases it may be convenient to use other norms. For instance an ellipsoidal norm \( ||s||_D = ||Ds|| \). For a complete discussion the interested reader may want to see the work of Moré [16].
2.1 The general trust region algorithm for (UCMIN)

We first specify a general trust region algorithm for the unconstrained minimization problem

**Algorithm 2.1** A general trust region algorithm for (UCMIN).

Given $x_0 \in \mathbb{R}^n$ and $\delta_0 > 0$

For $k = 0, 1, \cdots$, until convergence,

**step I.** (Compute a trial step.)

Determine a trial step $s_k$ as an approximate solution to subproblem (TRS).

**step II.** Evaluate the trial step $s_k$ and update the trust region radius.

If $s_k$ is accepted then

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

else set $x_{k+1} = x_k$

go to step I.

end if

**step III.** (Test for convergence.)

Now, let us discuss in detail each of these steps. It is necessary to define the solution $s$ of (TRS) and the algorithm for accepting the step and updating the trust region radius. If $x_c$ is the current iterate and $s$ is the solution of (TRS) let us define the quantities

**Actual reduction:**

$$Ared_c(s) = f(x_c) - f(x_c + s), \quad (2.1)$$

**Predicted reduction:**

$$Pred_c(s) = f(x_c) - q_c(s)$$

$$= q_c(0) - q_c(s). \quad (2.2)$$

The ratio between the actual reduction and the predicted reduction is used to decide whether the trial step is acceptable and to adjust the new trust region radius. If the step leads to sufficient decrease in the objective function, then the step is accepted and a new iterate is defined as $x_+ = x_c + s_c$. Otherwise the step is rejected,
the trust region radius reduced and a new step is computed by using the same model \( q_c(s) \).

Let us present a general scheme for evaluating the step and updating the trust region radius for a trust region method for unconstrained minimization.

**Algorithm 2.2**

**Evaluating the step and updating the trust region radius**

Given the constants: \( 0 < \alpha_1 < 1, \alpha_2 > 1, \ 0 < \eta_1 < \eta_2 < 1 \) and \( \delta_c \leq \delta_{\text{max}} \).

While \( \frac{\Delta r_{\text{red}}}{\Delta r_{\text{red},c}} < \eta_1 \)

Do not accept the step.

Reduce the trust region radius: \( \delta_c \leftarrow \alpha_1 \| s_c \| \), and compute a new step \( s_c \).

If \( \eta_1 \leq \frac{\Delta r_{\text{red}}}{\Delta r_{\text{red},c}} \leq \eta_2 \) then

Accept the step: \( x_+ = x_c + s_c \).

Keep the previous trust region radius: \( \delta_+ = \delta_c \).

end if

If \( \frac{\Delta r_{\text{red}}}{\Delta r_{\text{red},c}} > \eta_2 \) then

Accept the step: \( x_+ = x_c + s_c \).

Increase the trust region radius:

\[ \delta_+ = \min\{\delta_{\text{max}}, \max\{\delta_c, \alpha_2 \delta_c\}\} \].

end if

Since finding an accurate solution to (TRS) might be expensive, usually the conditions imposed on the step \( s_c \) are very mild. The step is required to satisfy a fraction of Cauchy decrease or FCD condition. This means \( s_c \) must predict via the quadratic model function at least as much as a fraction of the decrease given by the Cauchy step, that is, there exists a constant \( \sigma > 0 \) such that

\[
q_c(s_c) - q_c(0) \leq \sigma [q_c(s_c^{CP}) - q_c(0)],
\]

where

\[
s_c^{CP} = -l_c^{CP} \nabla f_c
\]
and its steplength

\[ t_c^{cp} = \begin{cases} \frac{\|\nabla f_c\|^2}{\nabla f_c^T G_c \nabla f_c} & \text{if } \frac{\|\nabla f_c\|^2}{\nabla f_c^T G_c \nabla f_c} \leq \delta_c \\ \frac{\delta_c}{\|\nabla f_c\|} & \text{otherwise.} \end{cases} \]

Thus, \( s_c^{cp} \) is the steepest descent step for \( q_c \) inside the trust region.

One of the results needed to prove convergence is a technical lemma which expresses (2.3) in a workable form.

**Lemma 2.1**

In the \( \ell_2 \)-norm, if the trial step satisfies a FCD-condition, then

\[ q_c(0) - q_c(s_c) \geq \frac{\sigma}{2} \|\nabla f_c\| \min \left\{ \frac{\|\nabla f_c\|}{\|G_c\|}, \delta_c \right\}. \quad (2.4) \]

**Proof**

See Powell [22], More [16] or Carter [2].

In the next section we will see that the global convergence analysis for trust region algorithms depends on the fact that the predicted reduction satisfies the weaker condition (2.4). Therefore, instead of solving (TRS) exactly we can compute a trial step that satisfies (2.4). The step \( s_c \) can be computed by dogleg-type techniques, see Powell [21] and Dennis and Mei [4] or by searching in the two-dimensional space spanned by the steepest descent direction and the Newton’s step, see Shultz, Schnabel and Byrd [24]. The subproblem (TRS) also can be solved approximately by a conjugate direction method which can be viewed as a generalized dogleg technique, see Steihaug [26] and Toint [29]. This way to compute a trial step is the topic for the Section 2.3. We have now specified completely the trust region algorithm for the unconstrained case. The convergence results apply to Algorithm 2.1 under the assumptions that the step satisfies condition (2.3) and the trust region radius is updated by Algorithm 2.2.

### 2.2 Global convergence results

The first global convergence results for a trust region algorithm for the unconstrained case was established by Powell [22]. It is important to observe that the convergence properties of trust region methods do not depend on the Newton step, it is only necessary to assume that the step satisfies a sufficient decrease condition of the form (2.3).
Theorem 2.1
Let \( f : \mathbb{R}^n \to \mathbb{R} \) be continuously differentiable and bounded below on \( \mathbb{R}^n \) and \( \{x_k\} \) a sequence of iterates generated by Algorithm 2.1 then
\[
\liminf_{k \to \infty} \|\nabla f_k\| = 0.
\]

Proof
See Powell [22] or Moré [16]. □

The key tool used in the proof of Theorem 2.1 is the inequality (2.4) given by Lemma 2.1. It would be desirable to replace \( \liminf \) by \( \lim \) in Theorem 2.1. This improvement can be obtained under the assumption that \( \nabla f(x) \) is uniformly continuous.

Theorem 2.2
If \( f : \mathbb{R}^n \to \mathbb{R} \) is bounded below on \( \mathbb{R}^n \), the sequence \( \{\|G_k\|\} \) bounded and \( \nabla f(x) \) is uniformly continuous then
\[
\lim_{k \to \infty} \|\nabla f_k\| = 0.
\]

Proof
See Thomas [27] or Moré [16]. □

Another attractive feature of the trust region algorithms is that it is possible to prove that
\[
\limsup_{k \to \infty} \mu_1[\nabla^2 f(x_k)] \geq 0,
\]
where \( \mu_1[A] \) denotes the smallest eigenvalue of the symmetric matrix \( A \). This result can be obtained assuming that \( f \) is twice continuously differentiable, that the matrix \( G_c \) is the exact Hessian \( \nabla^2 f(x_c) \), and that the step satisfies a fraction of optimal decrease or FOD condition. That is, if \( x_c \) is the current iterate, then the trial step \( s_c \) must predict via the quadratic model function at least a fraction of as much decrease as the Levenberg-Marquardt step does, that is there exists a constant \( \tau > 0 \) such that
\[
q_c(s_c) - q_c(0) \leq \tau[q_c(s_c^\text{LM}) - q_c(0)],
\]
where \( s_c^\text{LM} \) solves the trust region subproblem
\[
\begin{cases}
\text{minimize} & q_c(s) \\
\text{subject to} & \|s\| \leq \delta_c.
\end{cases}
\]
Theorem 2.3
If \( f: \mathbb{R}^n \rightarrow \mathbb{R} \) is twice continuously differentiable and bounded below on \( \mathbb{R}^n \) and \( \nabla^2 f(x) \) is bounded on the level set
\[
\{ x \in \mathbb{R}^n : f(x) \leq f(x_0) \}.
\]

If \( \{x_k\} \) is the sequence generated by Algorithm 2.1 under the assumption that the model function \( q_k(s) \) uses the exact Hessian matrix \( \nabla^2 f(x) \) and that the step \( s_k \) satisfies the condition (2.5), then:

i) The sequence \( \{\nabla f_k\} \) converges to zero.

ii) If \( \{x_k\} \) is bounded then there is a limit point \( x_* \) with \( \nabla^2 f(x_*) \) positive semidefinite.

iii) If \( x_* \) is an isolated limit point of \( \{x_k\} \), then \( \nabla^2 f(x_*) \) is positive semidefinite.

iv) If \( \nabla^2 f(x_*) \) is nonsingular for some limit point \( x_* \) of \( \{x_k\} \), then
   a) \( \nabla^2 f(x_*) \) is positive definite.
   b) \( \{x_k\} \) converges to \( x_* \).
   c) all the iterations are eventually successful, that is
      \[
      \frac{\text{Ared}_k}{\text{Pred}_k} \geq \eta_1
      \]
      and \( \{\delta_k\} \) is bounded away from zero.

Proof
See Fletcher [12], Sorensen [25], Moré and Sorensen [17] and Moré [16].

Shultz, Schnabel and Byrd [24] have obtained several variations on Theorem 2.3. The following variation is of interest because it does not require the full power of the fraction of optimal decrease condition, it suffices to assume that (2.3) holds.

Theorem 2.4 Let \( f: \mathbb{R}^n \rightarrow \mathbb{R} \) be twice continuously differentiable and bounded below on \( \mathbb{R}^n \) and assume that \( \nabla^2 f(x) \) is bounded below on the level set
\[
\{ x \in \mathbb{R}^n : f(x) \leq f(x_0) \}.
\]

Let \( \{x_k\} \) be the sequence generated by Algorithm 2.1 under the assumption that the model function \( q_k(s) \) uses the exact Hessian matrix \( \nabla^2 f(x) \)
and that the step $s_k$ satisfies the condition (2.3). If $\{x^*_k\}$ is a limit point of the sequence $\{x_k\}$ with $\nabla^2 f(x^*_k)$ positive definite then $\{x_k\}$ converges to $\{x^*_k\}$, all iterations are eventually successful and $\{\delta_k\}$ is bounded away from zero.

2.3 The Steihaug-Toint algorithm for unconstrained minimization

This section is devoted to describe the generalized dogleg algorithm introduced by Steihaug [26] and Toint [29], for solving approximately the trust region subproblem (TRS). This algorithm is based on the linear conjugate method and it is well-known for being suitable for large-scale problems. A discussion about conjugate direction algorithms can be found in Dennis and Turner [7].

The basic algorithm will terminate accordingly to the following three rules:

a) The algorithm will terminate if we have found a sufficiently good approximation to the Newton step, (step 3.). See [26].

b) If a descent direction of negative or zero curvature is found, the algorithm will take a step in that direction and to the boundary of the trust region, (step 1.).

c) If an iterate is outside the trust region, the trial step is a dogleg point defined as the intersection of the boundary of the trust region and the segment from the last iterate inside to the first iterate outside the trust region.

Algorithm 2.3
A modified conjugate gradient algorithm for (TRS)

Given $x_c$, the current iterate and the trust region of radius $\delta_c$.

step 0: (Initialization)

Set $\hat{s}_0 = 0$.
Set $r_0 = -(G_c \hat{s}_0 + \nabla f_c)$.
If $r_0 = 0$ then terminate.
Set $d_0 = r_0$.
Set $\hat{i} = 0$.

step 1: Compute $\gamma_i = d_i^T G_c d_i$.
If $\gamma_i > 0$ then go to step 2:
Otherwise (* $d_i$ is a direction of negative or zero curvature *)
compute $\tau > 0$ such that $\|\hat{s}_i + \tau d_i\| = \delta_c$.
Set $s_c = \hat{s}_i + \tau d_i$ and terminate.

**step 2:** Compute $\alpha_i = \frac{\|r_i\|^2}{\gamma_i}$.
Set $p_{i+1} = p_i + \alpha_i d_i$.
If $\|p_i\| < \delta_c$ go to **step 3**.
Otherwise (* the step is too long, take the dogleg step *)
compute $\tau > 0$ such that $\|\hat{s}_i + \tau d_i\| = \delta_c$.
Set $s_c = \hat{s}_i + \tau d_i$ and terminate.

**step 3:** Compute $r_{i+1} = r_i - \alpha_i G_c d_i$.

\[
\text{If } \frac{\|r_{i+1}\|}{\|\nabla f_c\|} \leq \xi_c, \quad (0 < \xi_c \leq \xi < 1) \quad \text{then} \quad (2.6)
\]

set $s_c = p_{i+1}$ and terminate.

**step 4:** Compute $\beta_i = \frac{\|r_{i+1}\|^2}{\|r_i\|^2}$.
Set $d_{i+1} = r_{i+1} + \beta_i d_i$.
Set $i = i + 1$ and go to **step 1**.

The framework of the Steihaug-Toint algorithm can be established by computing
the trial step $s_c$ in the general algorithm 2.1 by Algorithm 2.3.

**Algorithm 2.4**  The Steihaug-Toint algorithm for (UCMIN).

Given $x_0 \in \mathbb{R}^n$ and $\delta_0 > 0$
For $k = 0, 1, \cdots$, until convergence ,

**step I.** (Test for convergence.)

**step II.** (Compute a trial step.)

Determine a trial step $s_k$ as an approximate solution to subproblem
(TRS) by using Algorithm 2.3.

**step III.** Evaluate the trial step $s_k$ and update the trust region radius
by using Algorithm 2.2.

If $s_c$ is the solution of (TRS) obtained applying Algorithm 2.3, then

\[
s_c = \begin{cases} 
\hat{s}_{i+1} & \text{if (2.6) is used} \\
\hat{s}_i + \tau d_i & \text{if } \gamma_i > 0 \text{ and the step is too long} \\
\hat{s}_i + \tau d_i & \text{if } \gamma_i \leq 0.
\end{cases}
\]
Theorem 2.5
Let \( \hat{s}_j, j = 0, \ldots, i \) be the sequence of iterates generated by Algorithm 2.3. Then

i) \( q_c(\hat{s}_j), j = 0, \ldots, i \) is strictly decreasing and

\[
q_c(s_c) \leq q_c(\hat{s}_i). \tag{2.7}
\]

i) \( \|\hat{s}_j\|, j = 0, \ldots, i \) is strictly increasing and

\[
\|\hat{s}_c\| > \|\hat{s}_i\|. \tag{2.8}
\]

Proof
See Steihaug [26]. \qed

From Theorem 2.5, Steihaug [26] proves that

\[
q_k(0) - q_k(s_k) \geq \min \{ q_k(\tau \nabla f_k) : \|\tau \nabla f_k\| \leq \|s_k\| \} \tag{2.9}
\]

with equality only if \( i = 0 \) and the step is in the boundary of the trust region. Hence the step \( s_k \), obtained by applying Algorithm 2.3 satisfies the fraction of Cauchy decrease condition (2.4) with \( \beta = \frac{1}{2} \). Therefore Theorem 2.1 can be apply to Algorithm 2.4. That is

Theorem 2.6

If for all \( k = 0, 1, \ldots, \xi_k \leq \xi < 1 \), then

\[
\liminf_{k \to \infty} \|\nabla f_k\| = 0.
\]

Proof
See Steihaug [26]. \qed
Chapter 3

Trust region methods for constrained optimization

The objective of this research is to develop a global convergence theory for a whole class of algorithms. This family is characterized by the properties of the step that its members generate and because all of them evaluate the step by using the same merit function. This chapter presents all the pieces of a general nonlinear programming algorithm.

We begin by presenting a typical iteration of such a algorithm. In Section 3.2 we describe briefly the Successive Quadratic Programming (SQP) method. In Section 3.3 we describe some existing trust region subproblems. The condition that a general trial step must satisfy is the topic of Section 3.4. In Section 3.5 we discuss our choice of the merit function. The rule for evaluating the step and updating the trust region radius is the subject of Section 3.6. Then we discuss the rule for updating the penalty parameter and finally we state the algorithm.

3.1 The general trust region algorithm for (EQC)

If $x_c$ is the current iteration and $\delta_c > 0$ is the given trust region radius, a typical iteration for any nonlinear programming algorithm based on a trust region strategy that solves the equality constrained case is:

Algorithm 3.1 A typical TR-iteration.

Given $x_c$ and $\delta_c > 0$,

step I. Test for convergence.

step II. Compute a trial step:

Determine a trial step $s_c$ and set $x_+ = x_c + s_c$ where $s_c$ approximately solves a TR-subproblem.

step III. Update the penalty parameter in the merit function.
step IV. Evaluate the trial step $s_e$ and update the trust region radius.

Now, let us discuss in detail each of these steps.

3.2 The successive quadratic programming method

The algorithm that we propose is based on the SQP method. SQP can be described as follows: if $x_e$ is the current estimate to a solution $x_*$ of problem (EQC), the method finds a step $s_e \in \mathbb{R}^n$ by solving approximately the quadratic programming subproblem

$$\text{QP} \equiv \begin{cases} 
\text{minimize} & q_c(s) \\
\text{subject to} & \nabla C^T s + C_c = 0,
\end{cases}$$

where $q_c(s)$ is a quadratic model of the Lagrangian function associated with (EQC), $\ell(x, \lambda) = f(x) + \lambda^T C(x)$, at the current iterate $x_e$. We will denote the step $s_e$ by $s_{e\text{QP}}$ when the QP problem is solved exactly.

3.2.1 The quadratic model $q_c(s)$

We can find in the literature two different ways to construct a quadratic model of $\ell(x_e + s, \lambda)$ around $x_e$:

$$q_c(s) = \frac{1}{2} s^T H_c s + \nabla_x \ell^T_c s + \ell_c$$ (3.1)

or

$$q_c(s) = \frac{1}{2} s^T H_c s + \nabla f^T_c s + \ell_c.$$ (3.2)

If $s_e$ is the solution of this quadratic programming problem, the next estimate of $(x, \lambda)$ is defined as $x_+ = x_e + s_e$. If the model (3.1) is used we will obtain the correction $\Delta \lambda_c$, while if $\ell(x, \lambda)$ is modeled by (3.2), we will obtain directly the updated $\lambda_+$ for the Lagrange multiplier vector. Celis, Dennis and Tapia [3], El-Alem [8], [9], Powell and Yuan [23] and Williamson [32] use the model (3.1), while Vardi [31], Byrd, Schnabel and Shultz [1] and Omojokun [18] use (3.2).

In this work instead of considering the Lagrangian function as a function of $x$ and $\lambda$, we will use the multiplier substitution philosophy, suggested by Tapia [28]. It consists of treating the multiplier as a dependent variable rather than a parameter. That is, we substitute into $\ell$ the multiplier estimate written as a function of $x$, and
then we work with a problem in the variable $x$ alone. This will be convenient in the convergence analysis as we will see in the next chapters. Therefore we will be dealing with

$$f(x) = f(x) + \lambda(x)^T C(x)$$  \hspace{1cm} (3.3)

and its quadratic model around $x_c$ will be

$$q_c(s) = \frac{1}{2} s^T H_c s + \nabla C_c^T s + \ell_c.$$ \hspace{1cm} (3.4)

Under reasonable assumptions, if $H_c = \nabla^2 \ell_c$, the method converges $q$-quadratically and it converges $q$-superlinearly when $H_c$ is updated by an appropriate secant method.

In the implementation we will consider only the case $H_c = \nabla^2 \ell_c(x_c)$, although this is not crucial to our general approach.

The SQP method is one of the most popular approaches for solving nonlinear programming problems because under the standard assumptions it gives fast local convergence like Newton's method does. The drawbacks of the SQP method are:

1) It does not converge globally without modification.

2) The subproblem may not have a solution if $H_c$ is not positive definite on the null space of $\nabla C_c^T$.

Our approach will use a trust region strategy to handle these cases in a straightforward way because we know that algorithms for unconstrained problems based on the trust region strategy converge from poor starting guesses, and they can also cope well with directions of negative and zero curvature. The classical references for the topic are Levenberg [14], Marquardt [15], Heiden [13], Powell [22], Dennis and Mei [4], Sorenesen [25], Moré and Sorenesen [17], Moré [16], Steihaug [26] and Shultz, Schnabel and Byrd [24].

### 3.3 Existing trust region subproblems for (EQC)

The straightforward way to extend the trust region subproblem for (EQC) is:

Given $x_c$, find $s_c$ as solution of
\[
\begin{align*}
\text{minimize} \quad q_c(s) &= \frac{1}{2} s^T H_c s + \nabla_x \ell_c^T s + \ell_c \\
\text{subject to} \quad \nabla C_c^T s + C_c &= 0 \\
\|s\| &\leq \delta_c.
\end{align*}
\]

Observe that the trust region constraint and the linearized constraints may be inconsistent and thus the model subproblem will not have a solution. To overcome such a difficulty several subproblems have been proposed. For instance see Vardi [30] and [31], Celis, Dennis and Tapia [3], Byrd, Schnabel and Shultz [1], Omojokun [18], Powell and Yuan [23] and Williamson [32].

In this section we will describe the two main approaches existing in the literature for dealing with the case

\[
\{ s : \nabla C_c^T s + C_c = 0 \} \cap \{ s : \|s\| \leq \delta_c \} = \emptyset.
\]

They are:

1) the tangent space approach, and

2) the full space approach.

### 3.3.1 The tangent space approach

In the family of nonlinear programming algorithms based on the trust region strategy we can distinguish those algorithms in which the trial step is determined as

\[
s_c = s_c^n + s_c^t,
\]

where \( s_c^n \) is the normal component, that is \( s_c^n \) is inside of the trust region and in the normal direction of the null-space of the constraints, \( \mathcal{N}(\nabla C_c^T) \) and \( s_c^t \) is the component in the tangent space of the constraints given by

\[
s_c^t = W_c \bar{s}_c^t,
\]

with \( \bar{s}_c^t \in \mathbb{R}^{n-m} \) and \( W_c \) an \( n \times (n - m) \) matrix whose columns form a basis for \( \mathcal{N}(\nabla C_c^T) \).
At each iteration, the subproblem is transformed into an unconstrained trust region subproblem by moving onto the linear manifold $\nabla C^T_\epsilon s + C_\epsilon = 0$ along the normal from $x_\epsilon$ obtaining $s^n_\epsilon$ such that

$$\{ s : \nabla C^T_\epsilon (s - s^n_\epsilon) = 0 \} \cap \{ s : \|s\| \leq \delta_c \} \neq \emptyset.$$ 

More details will be given later on $s^n_\epsilon$.

Then the tangential component $s^t_\epsilon$ of the trial step can be computed by any of the existing algorithms for the unconstrained trust region subproblem

$$(TSS) \equiv \begin{cases} 
\text{minimize} & q_\epsilon(S + s^n_\epsilon) \\
\text{subject to} & \nabla C^T_\epsilon S = 0 \\
& \|S\| \leq \tilde{\delta}_c,
\end{cases}$$

where $\tilde{\delta}_c = \sqrt{\delta_c^2 - \|s^n_\epsilon\|^2}$.

Algorithms based on the tangent space strategy cope quite well with directions of zero or negative curvature away from a solution. A typical example of the tangent space subproblem is the Vardi subproblem. Vardi [30] and [31] suggests relaxing the linear constraint by replacing $C_\epsilon$ by $\alpha C_\epsilon$, where $\alpha \in [0, 1]$, is chosen to ensure that the (TSS) problem be feasible. See Figure 3.1. Thus,

$$s^n_\epsilon = -\alpha (\nabla C^T_\epsilon)^+ C_\epsilon,$$

where $(\nabla C^T_\epsilon)^+$ denote the pseudoinverse matrix of $\nabla C^T_\epsilon$. Observe that if $\alpha = 0$ then $\nabla C^T_\epsilon s + \alpha C_\epsilon = 0$ contains $s = 0$ and

$$\{ s : \nabla C^T_\epsilon s + C_\epsilon = 0 \} \cap \{ s : \|s\| \leq \delta_c \} \neq \emptyset.$$ 

Then, by a Vardi subproblem we mean:

$$(\text{VARDI}) \equiv \begin{cases} 
\text{minimize} & q_\epsilon(s) \\
\text{subject to} & \nabla C^T_\epsilon s + \alpha C_\epsilon = 0, \quad 0 \leq \alpha \leq 1 \\
& \|s\| \leq \delta_c.
\end{cases}$$

The drawback of the Vardi subproblem is that the step depends on the parameter $\alpha$ which it is not clear how to determine.
Figure 3.1 Vardi subproblem.

In 1989, Omojokun [18], used this approach to compute a trial step that does not depend on $\alpha$ as follows,

$$\begin{cases}
\text{minimize} & q_c(s) \\
\text{subject to} & \nabla C_c^T s = \nabla C_c^T s_c^{PY} \\
 & \|s\| \leq \delta_c
\end{cases}$$

where $s_c^{PY}$ solves the following problem

$$\begin{cases}
\text{minimize} & \frac{1}{2}\|\nabla C_c^T s + C_c\|^2 \\
\text{subject to} & \|s\| \leq \sigma \delta_c
\end{cases}$$

where $0 < \sigma < 1$. See Figure 3.2. In this case,

$$s_c^n = -(\nabla C_c^T)^+ \nabla C_c^T s_c^{PY}.$$ 

Unfortunately, to determine $s_c^{PY}$ may be computationally expensive.
In the next chapter we will discuss two techniques for computing $s^n_c$, for the algorithm that we are going to suggest. That algorithm can be viewed as an extension of the Steihaug-Toint algorithm for the unconstrained case to the equality constrained case.

![Figure 3.2 Powell-Yuan subproblem.](image)

3.3.2 The full space approach

The other approach to overcoming the problem of inconsistency is the full space strategy. Algorithms based on this approach instead of considering the decomposition of the trial step, compute $s_c$ at once in the whole space $\mathbb{R}^n$.

The classical example of this category of trust region subproblem is the well-known CDT subproblem proposed by Celis, Dennis and Tapia, 1984, see [3]. Instead of considering the linearized constraint $\nabla C_c^T s + C_c = 0$, they replace it by a very particular inequality: $\|\nabla C_c^T s + C_c\| \leq \theta_c$.

The key to the subproblem (and its variants) is the choice of $\theta_c$. Hence, by a CDT-
The subproblem we mean the following subproblem:

\[
(CDT) \equiv \begin{cases} 
\text{minimize} & q_c(s) \\
\text{subject to} & \|\nabla C_c^T s + C_c\| \leq \theta_c \\
& \|s\| \leq \delta_c,
\end{cases}
\]

where \(\theta_c \in \mathbb{R}\) is chosen as \(\theta_c = \|\nabla C_c^T \hat{s} + C_c\|\) for some \(\hat{s}\) such that \(\|\hat{s}\| \leq \delta_c\). See Figure 3.3.

There are several choices for the parameter \(\theta_c\). CDT chooses \(\theta_c\) based on a fraction of Cauchy decrease condition on \(\|\nabla C_c^T s + C_c\|\),

\[
\theta_c = \theta^{Cd}_c \equiv r\|\nabla C_c^T s_c \text{CP} + C_c\| + (1 - r)\|C_c\|
\]
where $0 < r \leq 1$ and $s_c^{CP}$ solves the problem,

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| \nabla C_c^T s + C_c \|^2 \\
\text{subject to} & \quad \| s \| \leq \delta_c \\
& \quad s = -t \nabla C_c C_c, \quad t \geq 0.
\end{align*}
\]

Note that in this case the CDT subproblem minimizes the quadratic model of \( \ell \) over the set of steps inside the trust region that gives at least \( r \) times as much decrease in the \( \ell_2 \)-norm of the residual of the linearized constraints as the Cauchy step. See Figure 3.3.

Now assume that the Cauchy point is the most feasible point for the subproblem and in addition it lies on the boundary of the trust region. For this reason, in order to prevent the possibility of a single point for the subproblem and obtain a meaningful trust region subproblem, CDT suggests \( r < 1 \), for instance \( r = 0.8 \).

Powell and Yuan in 1986 [23] consider a different choice of \( \theta_c \) :

\[
\theta_c = \theta_c^{PY} \equiv \| \nabla C_c^T s_c^{PY} + C_c \|
\]

where $s_c^{PY}$ solves

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| \nabla C_c^T s + C_c \|^2 \\
\text{subject to} & \quad \| s \| \leq \sigma \delta_c
\end{align*}
\]

and $0 < \sigma < 1$. See Figure 3.2.

This choice of \( \theta_c \) is more expensive than \( \theta_c^{Cd} \) and it provides faster convergence to a solution of $C(x) = 0$. However faster convergence to a solution of $C(x) = 0$ does not mean faster convergence to a solution of (EQC!). The fact that $\theta_c^{PY} \leq \theta_c^{Cd}$ means less freedom for the quadratic function-\( q_c(s) \) to achieve optimality and more emphasis on feasibility. We believe that in early iterations feasibility is not the most crucial issue. An advantage of $\theta_c^{PY}$ occurs when $\{ s : \| s \| \leq \sigma \delta_c \} \cap \{ s : \nabla C_c^T s + C_c = 0 \} \neq \emptyset$ because in that case the $s_c^{PY}$ step would be automatically chosen.

The CDT approach is very robust but currently too expensive unless the quadratic model is convex. Therefore it is not a good idea to use it for large scale problems. Recently, Dennis, Tapia, Martínez and Williamson [6] proposed a less expensive subproblem, based on an idea of Byrd, Schnabel and Shultz [24] for the unconstrained case. In this subproblem, the iterate is restricted to lie on the plane containing the
Cauchy step \( C_p \) and the \( s^{qp}_p \)-step if it exists. This leads to the following subproblem

\[
\begin{aligned}
\text{minimize} & \quad q_c(s) \\
\text{subject to} & \quad \|\nabla C_e^T s + C_c\| \leq \theta_e \\
& \quad \|s\| \leq \delta_e \\
& \quad s \in \text{span}\{s^{p}_c, d\},
\end{aligned}
\]

where \( d = s^{qp}_p \) if it exists or a descent direction of nonpositive curvature on \( \mathcal{N}(\nabla C_e^T) \), if it does not. Williamson [32] studies this subproblem in detail.

We finish the section noting that even though a full space trust region subproblem does not compute separately the normal and tangential components of the step, for theoretical reasons it is necessary to identify them. See for instance El-Alem [9] and Powell-Yuan [23].

### 3.4 The general trial step

We will be more specific in Chapter 4, but for our convergence theory we only require that the trial step is determined as

\[
s_c = s^n_c + s^t_c,
\]

where \( s^n_c \) and \( s^t_c \) are as on page 16. We will require that the components \( s^n_c \) and \( s^t_c \) satisfy a FCD condition on appropriate model functions. We will require that the normal component gives at least as much decrease as \( -n^{cp}_c \nabla C_e C_e \) on the quadratic model of the linearized constraint, where the steplength is given by

\[
n^{cp}_c = \begin{cases} 
\frac{\|\nabla C_e C_e\|^2}{\|\nabla C_e^T \nabla C_e C_e\|^2} & \text{if } \frac{\|\nabla C_e C_e\|^2}{\|\nabla C_e^T \nabla C_e C_e\|^2} \leq r \delta_e \\
\frac{\delta_e}{\|\nabla C_e C_e\|} & \text{otherwise},
\end{cases}
\]

and \( 0 \leq r < 1 \). Thus, we could take \( s^n_c = -n^{cp}_c (\nabla C_e^T) + \nabla C_e C_e \).

Consider the level sets of \( \|\nabla C_e^T s + C_c\|^2 \) and the trust region of radius \( r \delta_e \). The intersection of the trust region with the ellipsoidal cylinder \( \|\nabla C_e^T s + C_c\|^2 \leq \theta^{cd}_e \), where \( \theta^{cd}_e = \| -n^{cp}_c \nabla C_e^T \nabla C_c C_c + C_c\|^2 \), is the set of steps that satisfy a fraction of Cauchy decrease condition,

\[
S_c = \{ s : \|s\| \leq r \delta_e \} \cap \{ s : \|\nabla C_e^T s + C_c\|^2 \leq \theta^{cd}_e \}.
\]
Now we pick some linear manifold $\mathcal{M}_c$ parallel to the null-space of the constraints,
\[
\mathcal{M}_c = \{ s : \nabla C_c^T s_c = \nabla C_c^T s'_c, s'_c \in S_c \}.
\]
Thus,
\[
\mathcal{M}_c \cap \{ s : \| s \| \leq \delta_c \} = \{ s : \| s \| \leq \bar{\delta}_c \} \neq \emptyset,
\]
where $\bar{\delta}_c = \sqrt{1 - \| s_c^* \|^2}$, $\delta_c \geq \sqrt{1 - r^2 \delta_c}$, and $\bar{\delta}_c$ is the trust region radius of $\delta_c$, ensuring that $\mathcal{M}_c$ lies too near the boundary.
of the trust region of radius \( \delta_c \). Let \( s^n_c \) be the normal to \( M_c \). On the manifold \( M_c \), we consider the quadratic model \( q_c(s) \) of the Lagrangian function associated with the (EQC) problem. Then we ask the tangential component \( s^t_c \) to satisfy a fraction of Cauchy decrease condition from \( s^n_c \) on \( q_c(s) \) reduced to \( M_c \). See Figure 3.5. That is \( s^n_c + s^t_c \in G \cap M_c \), where

\[
G_c = \{ s^n_c + s : ||s|| \leq \delta_c, q_c(s^n_c + s) - q_c(s^n_c) \leq \sigma[q_c(s^n_c - t^C_p \nabla q_c(s^n_c)) - q_c(s^n_c)], \text{ for some } 0 < \sigma \leq 1 \}
\]

and

\[
t^C_p = \begin{cases} \frac{||\nabla q_c(s^n_c)||^2}{\nabla q_c(s^n_c)^T H_c \nabla q_c(s^n_c)} & \text{if } \frac{||\nabla q_c(s^n_c)||^2}{\nabla q_c(s^n_c)^T H_c \nabla q_c(s^n_c)} \leq \delta_c \\ \frac{\delta_c}{||\nabla q_c(s^n_c)||} & \text{otherwise.} \end{cases}
\]

**Figure 3.5** Fraction of Cauchy decrease on the \( \mathcal{N}(\nabla C^T_c) \).
We will refer to the set $\mathcal{G}_c$ as the **Gold region** on $\mathcal{M}_c$. Note that in the figure 3.5 for sake of simplicity we have drawn $q_c(s)$ convex. However if the iterates are far away from the solution we do not expect that the second order sufficient condition holds at $x_c$. See Moré [16].

### 3.5 The choice of the merit function

Our objective is to develop a general theory based on a trust region strategy for the optimization problem (EQC). Thus, if $x_c$ is the current iterate, we defined a local model problem around $x_c$, requirements on the approximate solution $s_c$, and now we need a procedure for accepting the step and updating the trust region radius.

We must decide if a trial step chosen to satisfy $s^n_c \in S_c$ and $s = s^n_c + s^1_c \in \mathcal{G}_c \cap \mathcal{M}_c$ is a **good step**, that is, if the step $s_c$ gives a new iterate $x_+$ that is a better approximation than $x_c$ to a solution of (EQC), say $x_*$. In constrained optimization better approximation should mean improvement not only in $f$ but also in the constraint error $\|C\|_2$. The evaluation of the trial step requires the choice of a merit function, which usually is a weighted combination of the objective function and the constraint violations.

In our research we use the augmented Lagrangian as the merit function and for the multiplier we use the Lagrange multiplier substitution method.

\[
\mathcal{L}(x; \rho) = f(x) + \lambda(x)^T C'(x) + \frac{1}{2} \rho C'(x)^T C(x), \quad \rho \geq 0
\]

\[
\lambda(x) = -\nabla C'(x)^T \nabla C(x) - \nabla C'(x)^T \nabla f(x).
\]

Since we choose the least square approximation formula we have the well-known exact penalty function proposed by Fletcher in 1972 [11]. This function has also been used as merit function by Powell and Yuan [23]. Note that the merit function is used only for evaluating the step and it is not related to the strategies or algorithms used for finding it.

### 3.6 Evaluating the trial step

Once we have solved the trust region subproblem for the trial step $s_c$, we have to decide if $s_c$ produces sufficient improvement in the merit function.

The quadratic model of $\mathcal{L}_c$
If \( x_c \) is the current iterate, let \( Q(s; \rho) \) be the quadratic model of \( L(x_c + s; \rho) \).

\[
Q_c(s; \rho) = L_c + \nabla L_c^T s + \frac{1}{2} s^T H_c s
\]

where

\[
\nabla L_c(\rho) = \nabla \ell_c + \rho \nabla C_c
= \nabla f_c + \nabla C_c \lambda_c + \nabla \lambda_c C_c + \rho \nabla C_c C_c.
\]

Since

\[
\nabla^2 L_c(\rho) = \nabla^2 \ell_c + \frac{1}{2} \rho [\nabla C_c \nabla C_c^T + \sum_{i=1}^m C_i(x_c) \nabla^2 C_i(x_c)]
= \nabla^2 f_c + \sum_{i=1}^m C_i(x_c) \nabla^2 \lambda_i(x_c) + \nabla \lambda_c \nabla C_c^T + \sum_{i=1}^m \lambda_i(x_c) \nabla^2 C_i(x_c) + \nabla C_c \nabla \lambda_c^T + \frac{1}{2} \rho [\sum_{i=1}^m C_i(x_c) \nabla^2 C_i(x_c) + \nabla C_c \nabla C_c^T],
\]

we approximate the Hessian matrix of \( L(x; \rho) \) by ignoring the terms that will vanish on the feasible region. Thus, we take our model Hessian to be

\[
H_c = \nabla^2 f_c + \sum_{i=1}^m \lambda_i(x_c) \nabla^2 C_i(x_c) + \nabla C_c \nabla \lambda_c^T + \nabla \lambda_c \nabla C_c^T + \frac{1}{2} \rho \nabla C_c \nabla C_c^T
\]

and the quadratic model of \( L(c_c + s; \rho) \) to be

\[
Q_c(s; \rho) = f_c + \lambda_c^T C_c + \frac{\rho}{2} ||C_c||^2 + (\nabla \ell_c + \rho \nabla C_c)^T s + \frac{1}{2} s^T (H_c + \rho \nabla C_c \nabla C_c^T) s
= \ell_c + \nabla \ell_c^T s + \frac{1}{2} s^T H_c s + \frac{\rho}{2} [||C_c||^2 + 2(\nabla C_c C_c^T s + \nabla C_c^T s)^2]
= q_c(s) + \frac{\rho}{2} ||\nabla C_c^T s + C_c||^2.
\]

If \( s_c \) is a trial step, to measure improvement we compare the actual reduction and predicted reduction from the current iterate \( x_c \) to the new one \( x_+ = x_c + s_c \).

**Actual reduction**

\[
Ared_c(s_c; \rho) = L(x_c; \rho) - L(x_+; \rho)
= L(x_c; \rho) - L(x_c + s_c; \rho)
= \ell_c - \ell_+ + \frac{\rho}{2} (||C_c||^2 - ||C_+||^2).
\]

(3.7)
Predicted reduction

\[ Pred_c(s_c; \rho) = L(x_c; \rho) - Q(s_c; \rho) \]
\[ = q_c(0) - q_c(s_c) + \frac{\rho}{2}(||C_c||^2 - ||\nabla C^T_c s_c + C_c||^2) \]
\[ = q_c(0) - q_c(s_c) + \frac{\rho}{2}(||C_c||^2 - ||\Theta||^2) \]
\[ = Q(0; \rho) - Q(s_c; \rho). \]  

(3.8)

It is important to note that other researchers use different definitions of predicted reduction. To signal this, we use for example \( Pred_c^{PY} \) to denote the Powell-Yuan form for \( Pred_c(s_c; \rho) \).

**Definition 3.1 (Successful Iteration)**

The current iteration is said to be successful if the improvement in \( L(x; \rho) \) is a sufficient proportion of that predicted by the model, that is

\[ \frac{\text{Arm}_{s_c; \rho}}{Pred_{s_c; \rho}} \geq \eta_1 \]

where \( \eta_1 \in (0, 1) \) is a fixed constant. Otherwise the iteration is said to be unsuccessful. A typical value for \( \eta_1 \) might be \( 10^{-4} \).

The strategy that we follow for accepting or rejecting the step, and updating the trust region radius is based on the standard rules for the unconstrained case. More details can be found in the book of Dennis and Schnabel [5]. However in order to establish our global convergence theory, we must introduce a modification in the strategy for updating the trust region radius.

At the beginning we set constants \( \delta_{\text{max}} > \delta_{\text{min}} \) and each time that we find an acceptable step we update the trust region radius to prepare for the next step. We compare the new value of \( \delta_c \) with \( \delta_{\text{min}} \) choosing the larger for the next trust region radius \( \delta_c \). In short, \( \delta_c \) can be reduced below \( \delta_{\text{min}} \) after an unacceptable step, but \( \delta_c \geq \delta_{\text{min}} \) must hold at the beginning of the next step after an acceptable step. Next, we present the scheme for evaluating the step and updating the trust region radius.

**Algorithm 3.2**

**Evaluating the step and updating the trust region radius**

Given the constants: \( 0 < \alpha_1 < 1, \alpha_2 > 1 \) and \( 0 < \eta_1 < \eta_2 < 1 \) and \( \delta_{\text{max}} > \delta_c \geq \delta_{\text{min}} \).
While $\frac{A_{red}}{P_{pred}} < \eta_1$ (e.g. $\eta_1 = 10^{-4}$)

Do not accept the step.
Reduce the trust region radius: $\delta_c \leftarrow \alpha_1\|s_c\|$ (e.g. $\alpha_1 = 0.5$), and
compute a new step $s_c$.

If $\eta_1 \leq \frac{A_{red}}{P_{pred}} \leq \eta_2$ (e.g. $\eta_2 = 0.5$) then
Accept the step: $x_+ = x_c + s_c$.
Set the trust region radius: $\delta_+ = \max\{\delta_c, \delta_{min}\}$.

end if

If $\frac{A_{red}}{P_{pred}} > \eta_2$ then
Accept the step: $x_+ = x_c + s_c$.
Increase the trust region radius:
$\delta_+ = \min\{\delta_{\max}, \max\{\delta_{\min}, \alpha_2\delta_c\}\}$ (e.g. $\alpha_2 = 2$).

end if

Observe that we have expressed the quantities $A_{red}$ and $P_{pred}$ as functions of $\rho$. Thus we will determine $\rho$ before evaluating the step. The right choice of the penalty parameter is one of the most important issues for algorithms that use the augmented Lagrangian as a merit function. So, the obvious questions are:

1) How is the penalty parameter updated? and,

2) Can we find an acceptable step?

The answer to the first question is the subject of the next section. A crucial aspect of the algorithms of the class that we are studying is precisely that property that we call the finite termination property: at each iterate an acceptable step must be found after solving finitely many trust region subproblems. This property allows us to say that an algorithm based on a trust region strategy is well defined. In fact, as we will see, this property is a direct consequence of the properties of the step and the scheme for updating the penalty parameter.

3.7 The penalty parameter

We have already chosen the merit function and defined $A_{red}$ and $P_{pred}$. From Definition 3.1 the evaluation of the trial step requires the determination of the penalty
parameter. Numerical experiences with nonlinear programming algorithms that use the augmented Lagrangian as merit function have shown that good performance of the algorithm depends on keeping the penalty parameter as small as possible. See Gill, Murray, Saunders and Wright [20] and Williamson [32]. Next we describe some rules for updating the penalty parameter.

3.7.1 Some existing schemes for updating the penalty parameter

The global convergence theories developed by El-Alem [8], [9] and Powell and Yuan [23], require that the sequence \( \{\rho_k\} \) be a nondecreasing sequence. El-Alem also requires that the predicted decrease in the merit function be at least as much as a fraction of Cauchy decrease in \( \|\nabla C_c^T s + C_c\|^2 \). See (2.3) and (2.4). Powell and Yuan require the predicted reduction be at least a fraction of optimal decrease, that is the step gives at least as much decrease as the Levenberg-Marquardt step on the quadratic model of the linearized error constraint. See (2.5). Then the idea is to establish a rule for updating the penalty parameter keeping it as small as possible.

It may be helpful to point out that \( \rho \) is updated at a different point in the algorithm than is usual. We are used to completing an iteration to obtain \( z_+ \) from \( x_c \), then, we obtain \( \delta_c \) from \( \delta_c \), compute new model information and start the next iteration. However, we update \( \rho \) after computing the trial step and before evaluating it. Thus, it is natural to think of updating \( \rho_+ \) to obtain \( \rho_+ \).

El-Alem scheme for updating the penalty parameter for the augmented Lagrangian

After computing \( s_c \), the trial step from the current iterate \( x_c \), El-Alem defines

\[
\rho_c^{EA} = \begin{cases} 
\rho_- & \text{if } Pred_c^{EA} \geq \rho_- \left(\|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2\right) \\
\rho_c^{EA} + \beta & \text{otherwise}
\end{cases}
\]

where

\[
Pred_c^{EA} = -\nabla f_c^T s_c - \frac{1}{2} s_c^T H_c s_c - \Delta \lambda_c^T (\nabla C_c^T s_c + C_c)
+ \rho_-(\|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2)
\]

\[
\rho_c^{EA} = 2 \frac{\nabla f_c^T s_c + \frac{1}{2} s_c^T H_c s_c + \Delta \lambda_c^T (\nabla C_c^T s_c + C_c)}{\|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2}
\]

and \( 0 < \beta < 1 \).
Powell and Yuan scheme for updating the penalty parameter for the augmented Lagrangian

The rule proposed by Powell and Yuan is as follows

\[
\rho_\text{PY}_c = \begin{cases} 
\rho_- & \text{if } \text{Pred}_\text{PY}_c \leq \frac{\rho_-}{2} \left( \|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2 \right) \\
2(\rho_- + \max\{0, \hat{\rho}_\text{PY}_c\}) & \text{otherwise}
\end{cases}
\]

where

\[
\text{Pred}_\text{PY}_c = -\nabla C_c^T s_c - \frac{1}{2} s_c^T H_c s_c - (\lambda_+ - \lambda_c)(\lambda_c - \frac{1}{2} \nabla C_c^T s_c + C_c)
\]
\[
+ \rho_c(\|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2)
\]

\[
\hat{s}_c = P(s_c - (I - \nabla C_c^T \nabla C_c)^{-1} \nabla C_c^T) s_c
\]

\[
\hat{\rho}_\text{PY}_c = 2 \frac{\text{Pred}_\text{PY}_c}{\|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2}.
\]

Omojokun scheme for updating the penalty parameter

Omojokun uses a modified \(\ell_2\)-penalty function

\[
\ell_2(x; \mu) = f(x) + \mu \|C(x)\|
\]

as merit function for a locally convergent nonlinear programming algorithm with

\[
\mu_c = \max\{\mu_-, \tilde{\mu}_c + \beta\}
\]

where

\[
\tilde{\mu}_c = \frac{-\frac{1}{2} s_c^T H_c s_c - \nabla f_c^T s_c}{\|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2}
\]

and \(0 < \beta < 1\).

3.7.2 The choice of the penalty parameter

Now we establish the rule for updating the penalty parameter in our algorithm. Observe that if \(s_c \in \text{IF}\), then \(\text{Pred}_c(s_c; \rho) = q_c(0) - q_c(s_c)\) does not depend on \(\rho\), so in this case we can keep the penalty parameter from the previous iteration.

Let us assume that \(s_c \notin \text{IF}\). For the augmented Lagrangian both El-Alem [8] and [9], and Powell and Yuan [23] keep the penalty parameter from the previous iteration.
if the decrease in the model of the merit function gives at least a fraction of the desired decrease in the quadratic model of the linearized constraints. This strategy aids in keeping the penalty parameter small.

At the current iterate the penalty parameter $\rho_c$ is increased if this is necessary to make

$$\text{Pred}_c(s_c; \rho) = \mathcal{Q}(0; \rho) - \mathcal{Q}(s_c; \rho)$$

$$= q_c(0) - q_c(s_c) + \frac{\rho}{2} (\|C_c\|^2 - \|\nabla C^T_c s_c + C_c\|^2)$$

$$> 0.$$ 

Thus we choose $\rho \geq \tilde{\rho}_c$ where

$$\tilde{\rho}_c = \frac{4[q_c(s_c) - q_c(0)]}{\|C_c\|^2 - \|\nabla C^T_c s_c + C_c\|^2}.\quad (3.9)$$

**Algorithm 3.3** Updating the penalty parameter

1. **Initialization**
   Set $\rho_0 = 1$
   and $\beta \in (0, 1)$.

2. **at the current iterate: $x_c$**
   Compute
   $$\text{Pred}_c(s_c; \rho) = q_c(0) - q_c(s_c) + \frac{\rho}{2} (\|C_c\|^2 - \|\nabla C^T_c s_c + C_c\|^2)$$
   
   **If** $\text{Pred}_c(s_c; \rho) \geq \frac{\rho}{4} (\|C_c\|^2 - \|\Theta_c\|^2)$ **then** $\rho_c = \rho -$
   **else** $\rho_c = \tilde{\rho}_c + \beta$,
   where
   $$\tilde{\rho}_c = \frac{4[q_c(s_c) - q_c(0)]}{\|C_c\|^2 - \|\nabla C^T_c s_c + C_c\|^2}.$$

**Lemma 3.1**

The sequence $\{\rho_k\}$ generated by the previous rule is nondecreasing.

**Proof**
Suppose that there exists an index $k$ such that

$$\rho_k > \rho_{k+1},\quad (3.10)$$
then
\[ \rho_k > \frac{4[q_{k+1}(s_{k+1}) - q_{k+1}(0)]}{\|C'_{k+1}\|^2 - \|\nabla C_{k+1}^T s_{k+1} + C_{k+1}\|^2 + \beta} \]

\[ \frac{\rho_k}{4}[\|C_{k+1}\|^2 - \|\Theta_{k+1}\|^2] > [q_{k+1}(s_{k+1}) - q_{k+1}(0)] + \frac{\beta}{4}[\|C_{k+1}\|^2 - \|\Theta_{k+1}\|^2] \]

then,

\[ Pred_{k+1}(s_{k+1}; \rho_k) > \left( \frac{\beta}{4} + \frac{\rho_k}{4} \right)[\|C'_{k+1}\|^2 - \|\Theta_{k+1}\|^2] \]

\[ \geq \frac{\rho_k}{4}[\|C'_{k+1}\|^2 - \|\Theta_{k+1}\|^2]. \quad (3.11) \]

Now, according with the rule for updating the penalty parameter, (3.11) implies that \( \rho_{k+1} = \rho_k \), which contradicts (3.10).

Finally we must make mention that El-Alem [10] has just suggested an algorithm and a rule for updating the penalty parameter that generate a non-monotonic sequence \( \{\rho_k\} \).

### 3.8 Termination of the algorithm

We use the first order necessary conditions for problem (EQC) to terminate the algorithm. The algorithm is terminated if the following criterion holds:

\[ \|W_r^T \ell_r\| + \|C'_{\ell}\| \leq \text{tol} \]

where \( 0 < \text{tol} \) is prespecified.

Observe that we are using the reduced gradient at \( x_e \) rather than the gradient of the Lagrangian. This is in keeping with our step decomposition into \( s^n_e \) toward feasibility and \( s^r_e \) which decreased the reduced quadratic.

### 3.9 Statement of the algorithm

Now that we have discussed the strategies for updating the penalty parameter and evaluating the step and updating the trust region radius we present a formal description of our nonlinear programming algorithm.

**Algorithm 3.4 The NLP-algorithm.**
step 0. (Initialization)
Given $x_0$, the initial guess.
Find $\lambda_0$, $\nabla \lambda_0$ and $W_0$.
Set $\delta_{\text{min}}$ and $\delta_{\text{max}}$.
Set $\rho_0 = 1$ and $\beta \in (0, 1)$.
Set $x_c = x_0$, the current iterate.

step 4. (Test for convergence)
Given tol > 0

If $\|W_c^T \nabla \ell(x_c)\| + \|C'(x_c)\| \leq \text{tol}$ then terminate.
else

Go to step 1.

end if

step 2. (Compute a trial step)

If $x_c$ is feasible then find a step $s_c$

that satisfies a fraction of Cauchy decrease condition on the
quadratic model $q_c(s)$ of the Lagrangian around $x_c$.

else (* $C'(x_c) \neq 0$ *)

a) Compute $s_c^e$ that satisfies a fraction of Cauchy decrease condi-
tion on the quadratic model of the linearized constraints.

b) Find $s_c^l$ that satisfies a fraction of Cauchy decrease condition
on the quadratic model $q_c(s_c^e + s)$ from $s_c^e$.

end if

step 3. (Update the penalty parameter)

Compute

$$\text{Pred}_c(s_c; \rho_-) = q_c(0) - q_c(s_c)$$
$$+ \frac{\rho_2}{2} \left[ \|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2 \right].$$

Update $\rho_-$ to obtain $\rho_c$ by using Algorithm 3.3

step 4. (Evaluate the step)

Compute

$$A\text{red}_c(s_c; \rho_c) = \ell_c - \ell_+ + \frac{\rho_c}{2} (\|C_c\|^2 - \|C_c\|^2)$$
Evaluate the step and update the trust region radius by using Algorithm 3.2.

If the step is accepted then go to step 1.
else
    go to step 2.
end if
Chapter 4

A trust region algorithm based on the tangent space approach

This chapter is concerned with proposing a particular step choice algorithm for step 1. of Algorithm 3.4. Section 4.1 gives an algorithm for solving the quadratic programming problem without a trust region constraint. It is based on conjugate directions. In Section 4.2, we will discuss how to select the vector $\Theta_c \in \mathbb{R}^m$. We will present two different ways for doing that accordingly to the dimension of the problem. Finally, we will state the complete algorithm for finding the trial step.

4.1 A conjugate direction algorithm for equality constrained quadratic programming

This part is concerned with developing an algorithm which is suitable for solving the large scale quadratic programming (QP)-problem:

\[
\begin{align*}
\text{(QP)} & \equiv \begin{cases} 
\text{minimize} & \frac{1}{2}s^T H s + h^T s \\
\text{subject to} & A s - b = 0
\end{cases}
\end{align*}
\]

where $h \in \mathbb{R}^n$, $H$ is symmetric and positive definite on $\mathcal{N}(A)$, $A \in \mathbb{R}^{m \times n}$ is large, and $m \leq n$. We solve the (QP)-subproblem as follows:

Consider $s_0$ such that $A s_0 - b = 0$. Then, starting from $s_0$, we minimize $q(s)$ on the manifold defined by the system of linear constraints by using a conjugate reduced-gradient algorithm (CRG).
4.1.1 The conjugate reduced-gradient algorithm

Let us begin by considering the problem of minimizing a quadratic function on the null space of a matrix $A$

\[
(QP) \equiv \begin{cases} 
\text{minimize } q(s) &= \frac{1}{2}s^T Hs + h^T s \\
\text{subject to } &A s = 0.
\end{cases}
\]

We will build a sequence of directions $\{p_1, \ldots, p_k, \ldots\}$ with $Ap_k = 0$ and a sequence of iterates $\{s_1, \ldots, s_k, \ldots\}$ such that $\{p_1, \ldots, p_k, \ldots\}$ is a set of conjugate directions with respect to $H$ restricted to $\mathcal{N}(A)$. Each $s_k$ is obtained by minimizing $q(s)$ on the subspace generated by $\{p_1, \ldots, p_k\}$. The sequence of directions is based on using the reduced gradient described by Wolfe in [33] in a conjugate direction algorithm.

The reduced problem

Consider the linear system $As = 0$. Since $A \in \mathbb{R}^{m \times n}$ and $\text{rank}(A) = m$, the columns of $A$ can be pivoted to be written as:

\[
A = [B|N] \quad \text{with} \quad B \in \mathbb{R}^{m \times m} \quad \text{nonsingular}.
\]

Then

\[
As = 0 \iff [B|N] \begin{bmatrix} s_B \\ s_N \end{bmatrix} = Bs_B + Ns_N = 0 \iff s_B = s_B(s_N) = -B^{-1}Ns_N.
\]

We choose $B$ so that $B^{-1}$ can be held in a sparse factored form or we can solve the system iteratively.

Now, looking at the Hessian of the quadratic function, let us permute the rows and columns so that we can consider the partition

\[
H = \begin{bmatrix} H_{11} & H_{12} \\
H_{21} & H_{22} \end{bmatrix}
\]

where:

- $H_{11}$ is a $m \times m$-matrix
- $H_{12}$ is a $m \times (n - m)$-matrix
- $H_{21} = H_{12}^T$ is a $(n - m) \times m$-matrix
- $H_{22}$ is a $(n - m) \times (n - m)$-matrix.
Thus, the quadratic function on $\mathcal{N}(A)$ can be expressed in terms of the non-basic variables $s_N$ as

$$q(s) = q(s_B, s_N) = \frac{1}{2} s_N^T \tilde{H} s_N + \tilde{h}^T s_N \equiv \tilde{q}(s_N),$$

where:

$$
\begin{align*}
\tilde{H} &= (B^{-1}N)^T H_{11}(B^{-1}N) - (B^{-1}N)^T H_{12} - H_{21}(B^{-1}N) + H_{22} \\
\tilde{h} &= -(B^{-1}N)^T h_B + h_N.
\end{align*}
$$

This leads us to make the following definitions:

**Definition 4.1 (Reduced function)**
The quadratic function $\mathcal{Q}$ is said to be the reduced function of $q$ on $\mathcal{N}(A)$.

**Definition 4.2 (Reduced Hessian)**
The matrix $\tilde{H}$ is said to be the reduced Hessian of $q$ on $\mathcal{N}(A)$.

Note that the reduced Hessian can be expressed as

$$\tilde{H} = W^T H W$$

where $W$ is the matrix:

$$W = \begin{bmatrix} -B^{-1}N \\ I_{n-m} \end{bmatrix}.$$ 

Similarly,

$$\tilde{h} = h_N + (-B^{-1}N)^T h_B = W^T h,$$

and the reduced function is

$$\tilde{q}(s_N) = \frac{1}{2} s_N^T W^T H W s_N + h^T W s_N.$$ 

This allows us to define a linear transformation from $\mathbb{R}^n$ into $\mathcal{N}(A)$ that associates with any $s \in \mathbb{R}^n$ the vector $W^T s$. The matrix $W$ defined above is said to be the **reducer matrix**.

**Definition 4.3 (Reduced gradient)**
The vector $W^T \nabla q(s) = \tilde{H} \tilde{s}_N + \tilde{h}$ is said to be the reduced gradient of $q$ on $\mathcal{N}(A)$ at $s = \hat{s}$, and $W^T r = -W^T \nabla q(\hat{s})$ is said to be the reduced residual on $\mathcal{N}(A)$ at $s = \hat{s}$. 
At this point, by using $W^T$, we have transformed the constrained problem into an unconstrained problem of dimension $n - m$

$$
\begin{align*}
\text{minimize} & \quad \bar{q}(s_N) = \frac{1}{2} s_N^T \bar{H} s_N + \bar{h}^T s_N \\
\text{subject to} & \quad W s_N \in \mathcal{N}(A).
\end{align*}
$$

Since the matrix $\bar{H} = W^T H W \in \mathbb{R}^{(n-m) \times (n-m)}$ is positive definite on $\mathcal{N}(A)$, we can find the unique minimizer of $\bar{q}$ by using the standard conjugate gradient (CG) algorithm.

Now it is straightforward to solve the problem

$$(QP) \equiv \begin{cases} 
\text{minimize} & q(s) = \frac{1}{2} s^T H s + h^T s \\
\text{subject to} & As = b, \quad b \neq 0
\end{cases}$$

under the assumptions that $\text{rank}(A) = m$ and $H$ is symmetric positive definite on $\mathcal{M} = \{s : As = b\}$. Let $s^f_c$ be a feasible point, i.e., $A s^f_c = b$. Then

$$\mathcal{M} = \{s : A(s - s^f_c) = 0\}.$$

If $s^f_c$ is a feasible point then the problem above can be transformed to

$$
\begin{cases} 
\text{minimize} & \bar{q}(S) \\
\text{subject to} & AS = 0,
\end{cases}
$$

where $S = s - s^f_c$ and

$$Q(S) = q(S + s^f_c) = \frac{1}{2} S^T H S + \nabla q(s^f_c)^T S + q(s^f_c).$$

Note that a feasible point can be found from

$$As = Bs_B + Ns_N = b \iff s_B = B^{-1}(b - Ns_N),$$

so choosing $s_N = 0$ and $s_B = B^{-1}b$, a feasible point is then

$$s^f_c = (s_B, s_N)^T = (B^{-1}b, 0)^T.$$

For the sake of simplicity let us call $s_N = \bar{s}$. Now let us establish the algorithm for finding a solution of problem $(QP)$ by using the reduced gradient idea.
Algorithm 4.1  **CRG-algorithm**

**step 0:** (Initialization)
- Set $\mathbf{s}_0 = 0$.
- Set $\mathbf{r}_0 = -\mathbf{W}^T (\mathbf{H}\mathbf{s}_0 + \mathbf{h})$.
- Set $\mathbf{d}_0 = \mathbf{r}_0$.
- Set $i = 0$.

**step 1:** (Determine the next iterate)
- Compute $\gamma_i = \mathbf{d}_i^T \mathbf{H}\mathbf{d}_i$.
- Compute $\alpha_i = \frac{\mathbf{r}_i^T \mathbf{d}_i}{\gamma_i}$.
- Set $\mathbf{s}_{i+1} = \mathbf{s}_i + \alpha_i \mathbf{d}_i$.

**step 2:** (Determine the next direction)
- Compute $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{H}\mathbf{d}_i$.
- If $\|\mathbf{r}_{i+1}\| \neq 0$ then
  - Compute $\beta_i = \frac{\|\mathbf{r}_{i+1}\|^2}{\|\mathbf{r}_i\|^2}$.
  - Set $\mathbf{d}_{i+1} = \mathbf{r}_{i+1} + \beta_i \mathbf{d}_i$.
  - Set $i = i + 1$ and go to step 1.
- otherwise set $\mathbf{s}_* = \mathbf{s}_{i+1}$ and terminate.

### 4.1.2 Properties of the reducer operator

Let us state some properties of the matrix $\mathbf{W}$.

**Proposition 4.1**
Let $\mathbf{\tilde{H}} = \mathbf{W}^T \mathbf{H} \mathbf{W}$ be the reduced Hessian. If $\{\mathbf{\tilde{p}}_1, \ldots, \mathbf{\tilde{p}}_k\}$ is a $\mathbf{\tilde{H}}$-conjugate set of vectors of $\mathbb{R}^{n-m}$, then $\{\mathbf{W}\mathbf{\tilde{p}}_1, \ldots, \mathbf{W}\mathbf{\tilde{p}}_k\}$ is a $\mathbf{H}$-conjugate set of vectors of $\mathbb{R}^n$.

**Proposition 4.2**
The columns of $\mathbf{W}$ form a basis for the null-space of the matrix $\mathbf{A}$.

**Proposition 4.3**
The column space of the matrix $\mathbf{W}$ coincides with the null-space of the constraints. That is, $\mathcal{R}(\mathbf{W}) = \mathcal{N}(\mathbf{A})$.

**Proposition 4.4**
The matrix $\mathbf{W}(\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T$ is the $l_2$-projection onto $\mathcal{N}(\mathbf{A})$.

The proofs of these properties are straightforward.
4.2 Choices for $\Theta_c$

![Figure 4.1](image)

**Figure 4.1** $\|s_c^{mn}\| \leq r \delta_c$.

Now we will discuss how to select the vector $\Theta_c \in \mathbb{R}^m$. We begin considering two regions, one of radius $\delta_c$ and the second of radius a fraction of $\delta_c$. We do this because we want the trust region subproblem to be meaningful and we are trying to avoid the possibility that $\{s : \nabla C_c^T s + C_c = \Theta_c\}$ lies too near the boundary of the trust region of radius $\delta_c$.

Let $s_c^{mn}$ be the solution of the minimum norm problem

$$(\text{MNP}) \equiv \begin{cases} 
\text{minimize} & \frac{1}{2} s^T s \\
\text{subject to} & \nabla C_c^T s + C_c = 0.
\end{cases}$$

If $s_c^{mn}$ is inside the trust region of radius $r \delta_c$ we take it as the linear feasible point and $\Theta_c = 0$, (see Figure 4.1). In this case $s_c^a = s_c^{mn}$. If the minimum norm solution is outside the trust region of radius $r \delta_c$, see Figure 4.2 and Figure 4.3, $\Theta_c$ needs to be modified to ensure that $\{s : \nabla C_c^T s + C_c = \Theta_c\} \cap \{s : \|s\| \leq r \delta_c\} \neq \emptyset$.

We can do this by using some of the following strategies

1) Craig's method.

2) Conjugate reduced gradient algorithm.
Figure 4.2 \( \| \phi_c^{mn} \| \geq \delta_c \).

Figure 4.3 \( r \delta_c < \| \phi_c^{mn} \| < \delta_c \).
4.2.1 Via conjugate reduced gradient algorithm

Observe that when we reduce the quadratic problem to an unconstrained problem on the tangent space of the constraints we can obtain cheaply a linear feasible point
\[ s_c^{\text{lf}} = (B_c^{-1}C_c, 0)^T. \]

Now since \( s_c^{\text{mn}} \) solves the quadratic problem (MNP) we can find it by using the CRG-algorithm starting from \( s_c^{\text{lf}} \). Observe that the Hessian matrix is the identity. Thus if the minimum norm step is larger than \( r\delta_c \) we compute the relaxed linear feasible step \( s_c^n \) on the boundary of the smaller trust region by moving the linear manifold given by the equation \( \nabla C_c^T s + C_c = 0 \) until it pass through \( s_c^n \). We believe that if the number of constraints is close to the number of variables, this is a convenient way to find \( s_c^n \).

4.2.2 Via Craig's method

The underdetermined linear system \( \nabla C_c^T s + C_c = 0 \) can be solved by using Craig's algorithm. It consists in making the transformation \( s = \nabla C_c y \) and applying the standard conjugate gradient algorithm to \( \nabla C_c^T \nabla C_c y + C_c = 0 \).

This strategy is based on the following result:

**Lemma 4.1** Let \( s^{\text{craig}} \) be the solution of applying Craig's algorithm to the underdetermined linear system \( As + b = 0 \). Then
\[ s^{\text{craig}} = s^{\text{mn}} = -A^T(AA^T)^{-1}b. \] (4.1)

**Proof**

From \( As = -b \) and the change of variable \( s = A^T y \), we have
\[ AA^T y = -b. \] (4.2)

The matrix \( AA^T \in \mathbb{R}^{m \times n} \) is symmetric and positive definite, then (4.2) has a unique solution,
\[ y^{\text{craig}} = -(AA^T)^{-1}b \] (4.3)

and
\[
    s^{\text{craig}} = A^T y^{\text{craig}} \\
    = -A^T(AA^T)^{-1}b \\
    = s^{\text{mn}}
\]
which is (4.2).

Therefore we can solve the minimum norm problem by applying a modified Craig algorithm. The algorithm will generate iterates until it finds the desired $s_c^{mn}$ or until it is aborted because it generates an iterate outside the smaller trust region. If this happens we make a dogleg as usual and move the linear manifold until it passes through that point, say $s_{\text{craig}}$. So $\Theta_c = \nabla C_c^T s_{\text{craig}} + C_c$. Then from that point and by using CRG/Steighaug-Toint algorithm we can find a relaxed linear feasible point, $s_c^n = \gamma s_c^{mn}$, $\gamma < 1$.

Observe that if $\|s_c^{mn}\| \leq r\delta_c$ by applying Craig's algorithm we can find $s_c^{mn}$ in no more than $m$ iterations.

Here is the algorithm.

Algorithm 4.2 Craig/Steighaug-Toint algorithm

Find $s_c^n$ by using Algorithm 2.3 starting with $s_0 = y_0 = 0$ and residual $r_0 = -C_c$.

Observe that when the Craig/Steighaug-Toint algorithm is applied, the algorithm will terminate because a step is too long has been found or because the termination rule (2.6) has been reached.

4.3 Computation of a trial step

At this point we have, an algorithm for solving the quadratic programming problem. We introduce the trust region idea by adding the trust region constraint to a relaxed linear constraint.

At the current iterate $x_c$ the trust region subproblem is defined by the model

\[
\begin{align*}
\text{minimize} & \quad q_c(s) \\
\text{subject to} & \quad \nabla C_c^T s + C_c = \Theta_c, \\
& \quad \|s\| \leq \delta_c,
\end{align*}
\]

where $\Theta_c$ is chosen in such a way that

\[
\{ s : \nabla C_c^T s + C_c = \Theta_c \} \cap \{ s : \|s\| \leq \delta_c \} \neq \emptyset. \tag{4.4}
\]
The trial step $s_c$ approximately solves
\[
\begin{align*}
\text{minimize} \quad & q_c(s) = \frac{1}{2} s^T H_c s + \nabla \ell_c^T s + \ell_c \\
\text{subject to} \quad & \nabla C_c^T (s - s_c^n) = 0 \\
& \|s\| \leq \delta_c.
\end{align*}
\]

Making the change of variable, $S = s - s_c^n$, we obtain
\[
\begin{align*}
\text{minimize} \quad & q_c(S + s_c^n) = \frac{1}{2} (S + s_c^n)^T H_c (S + s_c^n) + \nabla \ell_c^T (S + s_c^n) + \ell_c \\
\text{subject to} \quad & \nabla C_c^T S = 0 \\
& \|S + s_c^n\| \leq \delta_c.
\end{align*}
\]
or
\[
\begin{align*}
\text{minimize} \quad & Q_c(S) = q_c(S + s_c^n) = \frac{1}{2} S^T H_c S + \nabla q_c(s_c^n)^T S + q_c(s_c^n) \\
\text{subject to} \quad & \nabla C_c^T S = 0 \\
& \|S + s_c^n\| \leq \delta_c.
\end{align*}
\]

As we have pointed out in the previous section this subproblem can be reduced to
\[
\begin{align*}
\text{minimize} \quad & \hat{Q}_c(\bar{S}) = \frac{1}{2} \bar{S}^T W_c^T H_c W_c \bar{S} + \nabla q_c(s_c^n)^T W_c \bar{S} + q_c(s_c^n) \\
\text{subject to} \quad & \|\bar{S}\| \leq \bar{\delta}_c
\end{align*}
\]
where $\hat{Q}_c(\bar{S}) = Q_c(-B_c^{-1} N_c \bar{S}, \bar{S}) = Q_c(S_B, S_N) = Q_c(W_c \bar{S})$ with
\[
W_c = \begin{bmatrix}
-B_c^{-1} N_c \\
I_{n-m}
\end{bmatrix}
\]
and $\bar{\delta}_c = \sqrt{\delta_c^2 - \|s_c^n\|^2}$.

The algorithm that we propose for solving the trust region subproblem can be described as follows:
Since there exists $s_c^n$ such that $\|s_c^n\| \leq \delta_c$ and $\nabla C_c^T s_0 + C_c = \Theta_c$, we begin applying the CRG-algorithm to the reduced quadratic
\[
q_c(S + s_c^n)
\]
on $\mathcal{N}(\nabla C_c^T)$ by using the Algorithm 2.3. Here is the algorithm
Algorithm 4.3 CRG/Steinhaug-Toint algorithm

Find \( s_c \) by using Algorithm 2.3 staring with \( \tilde{s}_0 = s^n_c \) and residual \( r_0 = -W_c^T(H_c s^n_c + \nabla \ell_c) \).

4.4 Statement of the algorithm

Here we summarize the strategy for computing a trial step.

Algorithm 4.4

I. FEASIBILITY:
   1) If \( x_c \) is feasible go to II.
   2) Determine \( s_c^n \) by using Algorithm 4.1 starting from \( \tilde{s}_c^f \), where
      \( s_c^f = (-B_c^{-1}C_c, 0)^T \) or \( s_c^f = s_{\text{craig}} \) with \( \| s_{\text{craig}} \| = r\delta_c \).

II. MINIMIZATION:

   Find \( s_c \) by applying Algorithm 4.3 to

   \[
   \begin{align*}
   &\text{minimize } q_c(s) \\
   &\text{subject to} \\
   &\nabla C_c^T(s - s^n_c) = 0 \\
   &\| s \| \leq \delta_c.
   \end{align*}
   \]

For an "all-in-one-place" description of Algorithm 4.4 the interested reader is referred to Appendix A.

The following result is a direct consequence of the fact that we are using the CRG/Steinhaug-Toint algorithm.

Corollary 4.1

If \( \{ s_j, j = 1, \ldots, i \} \) \((i \leq n - m)\) is the sequence of iterates generated by Algorithm 4.4, then

i) \( \| \nabla C_c^T s_c + C_c \| = \| \nabla C_c^T s_j + C_c \| \quad \forall j = 1, \ldots, i. \)

ii) The sequence \( \{ \| s_j \| \} \quad j = 1, \ldots, i \) is strictly increasing and
    \[\| s_c \| > \| s_i \|.\]

iii) The sequence \( \{ q_c(s_j) \} \quad j = 1, \ldots, i \) is strictly decreasing and
    \[q_c(s_c) \leq q_c(s_i).\]
Proof

The first statement is obvious because we are minimizing $q_e(s)$ on $\mathcal{N}(\nabla C^T_e)$, and the last two are consequence of Theorem 2.5.
Chapter 5

Towards the global convergence theory

This chapter is concerned with the global convergence of the general nonlinear programming Algorithm 3.4. In the first section we state the assumptions under which the global convergence will be established. They are standard for algorithms for problem (EQC) based on trust region strategies. See El-Alem [9], and Byrd, Schnabel and Shultz [1]. In Section 5.2 we discuss the properties of the trial step which we write as $s_e = s^n_e + s^l_e$ where $s^n_e$ is normal to the manifold of feasible points for the linearized constraints and $\nabla C^T e s^l_e = 0$. The main issues are that the step predicts a fraction of Cauchy decrease on the quadratic model of the linearized constraints and that $s^l_e$ predicts fraction of Cauchy decrease on the quadratic model of the Lagrangian function associated with the problem. Finally, Section 5.3 deals with the predicted decrease in the quadratic model of the merit function.

5.1 The standard assumptions

We begin by stating the standard assumptions under which global convergence for the nonlinear programming Algorithm 3.4 will be proved.

Let us assume that the algorithm generates a sequence of iterates $\{x_k\}$. Then for such a sequence we assume,

A1. The sequence of iterates satisfies that for all $k$, $x_k$ and $x_k + s_k \in \mathbb{D}$, where $\mathbb{D}$ is a convex set of $\mathbb{R}^n$.

A2. $f, C \in C^2(\mathbb{D})$.

A3. $\text{rank}(\nabla C'(x)) = m$ for all $x \in \mathbb{D}$.

A4. $f(x), \nabla f(x), \nabla^2 f(x), C(x), \nabla C'(x), (\nabla C'(x)^T \nabla C'(x))^{-1}, \nabla^2 e_i(x)$ for $i = 1, \ldots, m$ are all uniformly bounded in $\mathbb{D}$.

This assumption means that for all $x \in \mathbb{D}$, there exist constants $\nu_0, \nu_1, \nu_2, \nu_3, \nu_4, \nu_5 > 0$ such that:
\[\|\nabla f(x)\| \leq \nu_0\]
\[\|\nabla^2 f(x)\| \leq \nu_1\]
\[\|C(x)\| \leq \nu_2\]
\[\|\nabla C(x)\| \leq \nu_3\]
\[\|\nabla C(x)^T \nabla C(x)^{-1}\| \leq \nu_4\]
\[\|\nabla^2 c_i(x)\| \leq \nu_5 \quad \forall i = 1, \ldots, m.\]

**Remark:**
Since we are using an approximation to the Hessian, which by the following lemma, results to be bounded we do not need to assume that the Hessian of the Lagrangian is bounded.

### 5.1.1 Properties

In this part we establish some consequences of the standard assumptions.

**Lemma 5.1**

Under the assumptions A1 - A4 the following hold.

i) \(\lambda(x) = -(\nabla C(x)^T \nabla C(x))^{-1} \nabla C(x)^T \nabla f(x)\) is bounded for all \(x \in \mathbb{D}\).

ii) The matrix

\[
\nabla \lambda(x) = -\left[ \sum_{i=1}^{m} e_i \nabla f(x) \right]^T \nabla^2 C_i(x) \\
\quad + \nabla C(x)^T \nabla^2 \ell(x, \lambda) \|_{(\nabla C(x)^T \nabla C(x))^{-1}}
\]

where \(e_i\) is the \(i^{th}\) coordinate vector, is bounded for all \(x \in \mathbb{D}\).

iii) \(\nabla \ell(x) = \nabla f(x) + \nabla C(x) \lambda(x) + \nabla \lambda(x) C(x)\) is bounded for all \(x \in \mathbb{D}\).

iv) \(H(x) = \nabla^2 f(x) + \nabla^2 C(x) \lambda(x) + \nabla C(x) \nabla \lambda(x)^T + \nabla \lambda(x) \nabla C(x)^T\) is bounded for all \(x \in \mathbb{D}\).

The matrix \(H(x)\) is an approximation of the Hessian of the Lagrangian, where the terms containing second order information that will vanish on the feasible region are ignored.
Proof

i) From the definition of $\lambda(x)$,

$$
\lambda(x) = -(\nabla C(x)^T \nabla C(x))^{-1} \nabla C(x)^T \nabla f(x),
$$

so

$$
\|\lambda(x)\| \leq \|\nabla C(x)^T \nabla C(x))^{-1}\| \|\nabla C(x)^T\| \|\nabla f(x)\| \leq \nu_4 \nu_2 \nu_0,
$$

which is bounded by using the standard assumption. Letting $K_0 = \nu_4 \nu_2 \nu_0$, we can write $\|\lambda(x)\| \leq K_0$.

ii) The derivative of $\ell(x, \lambda)$ with respect to the first variable $x$ is

$$
\nabla_x \ell(x, \lambda) = \nabla f(x) + \nabla C(x) \lambda
$$

$$
= \nabla f(x) - \nabla C(x)(\nabla C(x)^T \nabla C(x))^{-1} \nabla C(x)^T \nabla f(x)
$$

$$
= \mathcal{P}(x) \nabla f(x),
$$

where $\mathcal{P}(x) = I - \nabla C(x)(\nabla C(x)^T \nabla C(x))^{-1} \nabla C(x)^T$ is the projection matrix on the null space of the constraints.

Now we can write

$$
\nabla C(x)^T \nabla_x \ell(x, \lambda) = 0,
$$

which is equivalent to

$$
\nabla C_i(x)^T \nabla_x \ell(x, \lambda) = 0 \quad \forall i = 1, \ldots, m
$$

$$
\sum_{i=1}^{m} \epsilon_i (\nabla C_i(x)^T \nabla_x \ell(x, \lambda)) = 0. \quad (5.1)
$$

where $\epsilon_i \in \mathbb{R}^m$ is the $i^{th}$ coordinate vector.

For all $i = 1, \ldots, m$, let us define

$$
\phi_i(x) = \nabla C_i(x)^T \nabla_x \ell(x, \lambda).
$$

Then (5.1) becomes

$$
\sum_{i=1}^{m} \epsilon_i \phi_i(x) = 0.
$$

Computing the directional derivative, we obtain

$$
\sum_{i=1}^{m} \epsilon_i \phi_i'(x)(\eta) = 0.
$$
For all $i = 1, \ldots, m$, let us calculate $\phi'_i(x)(\eta)$, for all $\eta \in \mathbb{R}^n$, $\eta \neq 0$

\[
\phi'_i(x)(\eta) = [\nabla^2 C_i(x)(\eta)]^T \nabla_x \ell(x, \lambda) \\
+ \nabla C_i(x)^T [\nabla^2_{xx} \ell(x, \lambda) + \nabla C(x) \nabla \lambda(x)^T](\eta) \\
= \nabla \ell(x, \lambda)^T \nabla^2 C_i(x)(\eta) \\
+ \nabla C_i(x)^T \nabla^2_{xx} \ell(x, \lambda)(\eta) + \nabla C_i(x)^T \nabla C(x) \nabla \lambda(x)^T(\eta).
\]

Then for all $\eta \in \mathbb{R}^n$, $\eta \neq 0$,

\[
\sum_{i=1}^m e_i \phi'_i(x)(\eta) = \sum_{i=1}^m e_i [\nabla \ell(x, \lambda)^T \nabla^2 C_i(x)(\eta) \\
+ \nabla C_i(x)^T \nabla^2_{xx} \ell(x, \lambda)(\eta) + \nabla C_i(x)^T \nabla C(x) \nabla \lambda(x)^T(\eta)] \\
= \sum_{i=1}^m e_i [\nabla \ell(x, \lambda)^T \nabla^2 C_i(x) + \nabla C_i(x)^T \nabla^2_{xx} \ell(x, \lambda) \\
+ \nabla C_i(x)^T \nabla C(x) \nabla \lambda(x)^T](\eta) \\
= 0.
\]

Then, for any $\eta \in \mathbb{R}^n$, $\eta \neq 0$ we have

\[
\sum_{i=1}^m e_i [\nabla \ell(x, \lambda)^T \nabla^2 C_i(x)] + \sum_{i=1}^m e_i [\nabla C_i(x)^T \nabla^2_{xx} \ell(x, \lambda)] + \\
\sum_{i=1}^m e_i [\nabla C_i(x)^T \nabla C(x) \nabla \lambda(x)^T] = \\
\sum_{i=1}^m e_i [\nabla \ell(x, \lambda)^T \nabla^2 C_i(x)] + \\
\nabla C(x)^T \nabla^2_{xx} \ell(x, \lambda) + \nabla C(x)^T \nabla C(x) \nabla \lambda(x)^T = 0.
\]

Then,

\[
\nabla \lambda(x)^T = -(\nabla C(x)^T \nabla C(x))^{-1} \left[ \sum_{i=1}^m e_i \nabla \ell(x, \lambda)^T \nabla^2 C_i(x) \\
+ \nabla C(x)^T \nabla^2_{xx} \ell(x, \lambda) \right].
\] (5.2)

Now, let us find a bound for $\nabla \lambda(x)$:

\[
||\nabla \lambda(x)|| \leq ||(\nabla C(x)^T \nabla C(x))^{-1}|| \left[ \sum_{i=1}^m ||\nabla^2 C_i(x)|| \ ||\nabla \ell(x, \lambda)|| \\
+ ||\nabla C(x)^T|| \ ||\nabla^2_{xx} \ell(x, \lambda)|| \right].
\]
From the standard assumptions,
\[
\|\nabla_x \ell(x, \lambda)\| \leq \|\nabla f(x)\| + \|\nabla C(x)\| \|\lambda(x)\|
\leq \nu_0 + \nu_5 K_0.
\]
\[
\|\nabla^2_x \ell(x, \lambda)\| \leq \|\nabla^2 f(x)\| + \sum_{i=1}^m |\lambda_i(x)| \|\nabla^2 C_i(x)\|
\leq \nu_1 + \nu_5 \sum_{i=1}^m |\lambda_i(x)|
= \nu_1 + \nu_5 \|\lambda(x)\|_1
\leq \nu_1 + \nu_5 \sqrt{m} \|\lambda(x)\|_2
\leq \nu_1 + \nu_5 \sqrt{m} K_0.
\]

Then,
\[
\|\nabla \lambda(x)\| \leq \nu_4 \left(\nu_0 + \nu_5 K_0\right) \sum_{i=1}^m \nu_{5i} + \nu_5 (\nu_1 + \nu_5 \sqrt{m} K_0)
= K_1.
\]

iii) By differentiating \(\ell(x)\) we obtain
\[
\nabla \ell(x) = \nabla f(x) + \nabla C(x) \lambda(x) + \nabla \lambda(x) C(x)
\]
so
\[
\|\nabla \ell(x)\| \leq \|\nabla f(x)\| + \|\nabla C(x)\| \|\lambda(x)\| + \|\nabla \lambda(x)\| \|C(x)\|
\leq \nu_0 + \nu_1 \nu_2 + \nu_5 K_0 = K_2.
\]

iv) Using the definition of \(H(x)\) and the previous results, we have
\[
H(x) = \nabla^2 f(x) + \nabla^2 C(x) \lambda(x) + \nabla C(x) \nabla \lambda(x)^T + \nabla \lambda(x) \nabla C(x)^T
\]
\[
\|H(x)\| \leq \|\nabla^2 f(x)\| + \sum_{i=1}^m |\lambda_i(x)| \|\nabla^2 C_i(x)\| + \|\nabla C(x)\| \|\nabla \lambda(x)^T\| + \|\nabla \lambda(x)\| \|\nabla C(x)^T\|
\leq \nu_1 + \nu_5 \sqrt{m} K_0 + 2 \nu_5 K_1
= K_3.
\]

\(\square\)
Lemma 5.2

The reducer matrix

\[ W(x) = \begin{bmatrix} -B(x)^{-1}N(x) \\ I_{n-m} \end{bmatrix} \]

is bounded for all \( x \in \mathcal{ID} \).

**Proof**

For all \( x \in \mathcal{ID} \), \( \|\nabla C(x)^T\| = \| [B(x) \mid N(x)] \| \leq \nu_3 \). Clearly, \( B(x) \) and \( N(x) \) are bounded, that is, there exist constants \( \nu_B, \nu_N > 0 \) such that the Frobenius norms of \( B(x) \) and \( N(x) \) are bounded \( \|B(x)\|_F \leq \nu_B \) and \( \|N(x)\|_F \leq \nu_N \).

Now, let us compute:

\[ \|W(x)\|_F^2 = tr(W(x)W(x)^T) \]

\[ = tr \begin{bmatrix} (B(x)^{-1}N(x))^T B(x)^{-1}N(x) & (B(x)^{-1}N(x))^T \\ B(x)^{-1}N(x) & I_{n-m} \end{bmatrix} \]

\[ = tr((B(x)^{-1}N(x))^T B(x)^{-1}N(x)) + tr(I_{n-m}) \]

\[ = \|B(x)^{-1}N(x)\|_F^2 + n - m. \]

The continuity of \( B(x) \) for all \( x \in \mathcal{ID} \) and the well-known Banach Perturbation Lemma (for a reference, see the book of Ortega and Rheinboldt [19], page 46) allow us to write \( \|B(x)^{-1}\| \leq \nu_{B^{-1}} \), whence

\[ \|W(x)\|_F^2 \leq \nu_{B^{-1}}^2 \nu_N^2 + n - m \]

\[ \|W(x)\|_2 \leq \sqrt{\nu_{B^{-1}}^2 \nu_N^2 + n - m} \]

\[ = \omega. \]

(5.3)

\[ \Box \]

5.2 Properties of the trial step

In this section we deal with the properties of the trial step. Let us remember that it can be written in two pieces \( s_c = s_c^n + s_c^t \).

The following property of the normal component \( s_c^n \) of \( s_c \) is an immediate consequence of the standard assumptions.
Lemma 5.3
Let $s^n_e$ be the relaxed linear feasible step at the current iterate $x_e$, then under the standard assumptions

$$
\|s^n_e\| \leq K_4 \|C_e\| \tag{5.4}
$$

where $K_4 = \nu_3 \nu_4$.

Proof
If $x_e \in \mathbb{F}$, then $s^n_e = 0$ and so the result holds. If $x_e \notin \mathbb{F}$, then by the way we determine $s^n_e$, we have at each iteration that

$$
\|s^n_e\| \leq \|s^{mn}_e\|
= \|\nabla C_e (\nabla C^T_e \nabla C_e)^{-1} C_e\|
\leq \|\nabla C_e\| \|(\nabla C^T_e \nabla C_e)^{-1}\| \|C_e\|
\leq \nu_3 \nu_4 \|C_e\|
= K_4 \|C_e\|.
$$

Hence, the result is established. \hfill \square

Lemma 5.4
Let $s_e$ be the step generated by CRG/Steinhaug-Toint algorithm at the current iterate $x_e$. Then $s_e$ satisfies a FCD-condition on the quadratic model of the linearized constraints, i.e.

$$
\|\nabla C^T_e s_e + C_e\| \leq \|\nabla C^T_e s^{CP}_e + C_e\| \tag{5.5}
$$

where $s^{CP}_e$ solves the subproblem

$$
\begin{cases}
\text{minimize} & \|\nabla C^T_e s + C_e\| \\
\text{subject to} & \|s\| \leq \nu \delta_e \\
& s = -t \nabla C^T_e C_e, \quad t \geq 0.
\end{cases}
$$

Proof
Since

$$
\|\nabla C^T_e s_e + C_e\| = \|\nabla C^T_e s^{CP}_e + C_e\|
$$
to prove (5.5) it suffices to prove:

$$\|\nabla C^T s^u + C_c\| \leq \|\nabla C^T s^m C_c + C_c\|.$$  

Now, $s^u = \gamma s^m$, with $0 \leq \gamma \leq 1$. If $\gamma = 0$ then $x_c \in I$, $(\Theta = 0 \in IR^m)$, and it is obvious that the trial step satisfies (5.5).

If $\gamma = 1$ then $s^u = s^m$ $(\Theta = 0 \in IR^m)$ where $s^m$, the minimum norm solution to $\nabla C^T s + C_c = 0$ is the Levenberg-Marquardt step, that is it solves the problem

$$\begin{cases}
\text{minimize} & \frac{1}{2}\|\nabla C^T s + C_c\|^2 \\
\text{subject to} & \|s\| \leq r\delta_c,
\end{cases}$$

and it is clear that

$$\|\nabla C^T s^m + C_c\| \leq \|\nabla C^T s^m C_c + C_c\|.$$  

Now, if $0 < \gamma < 1$, note that $s^m$ is also the solution to the problem

$$\begin{cases}
\text{minimize} & \frac{1}{2}\|\nabla C^T s + C_c\|^2 \\
\text{subject to} & \|s\| \leq \|s^m\|.
\end{cases}$$

Then

$$\|\nabla C^T s^m + C_c\| \leq \|\nabla C^T C_c + C_c\|$$

where $C^T C_c$ solves

$$\begin{cases}
\text{minimize} & \frac{1}{2}\|\nabla C^T s + C_c\|^2 \\
\text{subject to} & \|s\| \leq \|s^m\| \\
s = -t\nabla C^T C_c, \quad t \geq 0.
\end{cases}$$

It is obvious that if

$$X = \{s : \|s\| \leq r\delta_c, s = -t\nabla C^T C_c, \quad t \geq 0\}$$

and

$$Y = \{s : \|s\| \leq \|s^m\|, s = -t\nabla C^T C_c, \quad t \geq 0\}$$

$X \subset Y$ and

$$\min\left\{\frac{1}{2}\|\nabla C^T s + C_c\|^2, s \in Y\right\} \leq \min\left\{\frac{1}{2}\|\nabla C^T s + C_c\|^2, s \in X\right\}$$
that is
\[ \|\nabla C_c^T s_c^{C_P} + C_c\| \leq \|\nabla C_c^T s_c^{C_P} + C_c\| \]

and
\[ \|\nabla C_c^T s_c^{mn} + C_c\| \leq \|\nabla C_c^T s_c^{C_P} + C_c\|. \]

Let us calculate
\[
\begin{align*}
\|\nabla C_c^T s_c^{mn} + C_c\| &= \|\nabla C_c^T \gamma s_c^{mn} + C_c\| \\
&= \|\nabla C_c^T \gamma s_c^{mn} + \gamma C_c + C_c - \gamma C_c\| \\
&= \|\gamma(\nabla C_c^T s_c^{mn} + C_c) + (1 - \gamma)C_c\| \\
&\leq \gamma \|\nabla C_c^T s_c^{mn} + C_c\| + (1 - \gamma)\|C_c\| \\
&\leq \gamma \|\nabla C_c^T s_c^{C_P} + C_c\| + (1 - \gamma)\|C_c\|
\end{align*}
\]

which is the desired result.

\[ \square \]

**Corollary 5.1**

Let \( s_c \) be the trial step at the current iteration. Under the standard assumptions there exist constants \( K_5 \) and \( K_6 \), which do not depend on \( p_e \), such that

\[ \|C_c\|^2 - \|\nabla C_c T s_c + C_c\|^2 \geq K_5 \|C_c\| \min\{K_6\|C_c\|, r \delta_c\}. \quad (5.6) \]

**Proof**

The proof is a modification of one given by El-Alem for a similar result, (Lemma 6.1 of [9]).

From Lemma 5.4, the step \( s_c \) is at least as linearly feasible as the Cauchy step \( s_c^{C_P} \):

\[ \|C_c\|^2 - \|\nabla C_c T s_c + C_c\|^2 \geq \|C_c\|^2 - \|\nabla C_c^T s_c^{C_P} + C_c\|^2. \]

The Cauchy step is \( s_c^{C_P} = -t_c^{C_P} \nabla C_c T C_r \) where

\[ t_c^{C_P} = \begin{cases} 
\frac{\|\nabla C_c T C_r\|}{\|\nabla C_c T C_c\|} & \text{if } \frac{\|\nabla C_c T C_r\|^2}{\|\nabla C_c T C_c\|^2} \leq r \delta_c \\
\frac{r \delta_c}{\|\nabla C_c T C_c\|} & \text{otherwise.}
\end{cases} \]
Let us suppose that the Cauchy point is inside the trust region of radius \( r\delta_c \), it means
\[
\frac{\|\nabla C_e^T C_e\|^3}{\|\nabla C_e^T \nabla C_e C_e\|^2} \leq r\delta_c.
\]

Then
\[
\|C_e\|^2 - \|\nabla C_e s_c + C_e\|^2 \geq -\|\nabla C_e^T s_c^p\|^2 - 2(\nabla C_e C_e^T)s_c^p \nabla C_e C_e\|
\]
\[
= -\|\nabla C_e^T \|\nabla C_e\|^2 \nabla C_e C_e\|^2
\]
\[
+ 2(\nabla C_e C_e^T) \frac{\|\nabla C_e C_e\|^2}{\|\nabla C_e^T \nabla C_e C_e\|^2} \nabla C_e C_e
\]
\[
= \frac{\|\nabla C_e C_e\|^4}{\|\nabla C_e^T \nabla C_e C_e\|^2}
\]
\[
= \frac{\|\nabla C_e C_e\|^2}{\|\nabla C_e^T \nabla C_e C_e\|^2} \|\nabla C_e C_e\|^2
\]
\[
\geq \frac{1}{\nu_3^2} \|\nabla C_e C_e\|^2. \tag{5.7}
\]

On the other hand if \( \|s_c^p\| \geq r\delta_c \), then we have
\[
\|C_e\|^2 - \|\nabla C_e s_c + C_e\|^2 \geq -(\nabla C_e^T s_c^p\|^2 - 2(\nabla C_e C_e^T)s_c^p \nabla C_e C_e\|
\]
\[
= -(\nabla C_e^T \delta_c \nabla C_e C_e\|^2 + 2(\nabla C_e C_e^T) \|\nabla C_e C_e\|)
\]
\[
= -(r\delta_c)^2 \frac{\|\nabla C_e^T \nabla C_e C_e\|^2}{\|\nabla C_e C_e\|^2} + 2r\delta_c \|\nabla C_e C_e\|
\]
\[
= r\delta_c \|\nabla C_e C_e\| \left[ 2 - r\delta_c \frac{\|\nabla C_e^T \nabla C_e C_e\|^2}{\|\nabla C_e C_e\|^3}\right]
\]
\[
\geq r\delta_c \|\nabla C_e C_e\|. \tag{5.8}
\]

From (5.7) and (5.8), we can write
\[
\|C_e\|^2 - \|\nabla C_e s_c + C_e\|^2 \geq \|\nabla C_e C_e\| \min \left\{ \frac{\|\nabla C_e C_e\|}{\nu_3^2}, r\delta_c \right\}. \tag{5.9}
\]

From inequality (4.4), we have \( \|\nabla C_e C_e\| \geq \frac{1}{K_4^2} \|C_e\| \), and defining
\[
K_5 = \frac{1}{K_4^2} \quad \text{and} \quad K_6 = \frac{1}{K_4 \nu_3^2}
\]
we obtain (5.6)

\[ \|C_c\|^2 - \|\nabla C_c s_c + C_c\|^2 \geq K_5\|C_c\| \min\{K_6\|C_c\|, r\delta_c\}. \]

This completes the proof. \qed

The following lemma shows that the null-space component \( s^n_c \), satisfies a fraction of Cauchy decrease condition on the quadratic model of the Lagrangian.

**Lemma 5.5**

Let \( s_c \) be a trial step generated by Algorithm 4.4. Then there exists a constant \( K_7 \), which does not depend on \( x_c \) such that

\[ q_c(s^n_c) - q_c(s_c) \geq \frac{1}{2}\|W_c^T \nabla q_c(s^n_c)\| \min\{\delta_c, K_7\|W_c^T \nabla q_c(s^n_c)\|\}. \tag{5.10} \]

**Proof**

Since we are solving the reduced problem

\[ \tilde{Q}_c(\tilde{s}) = \frac{1}{2} \tilde{s}^T W_c^T H_c W_c \tilde{s} + \nabla q_c(s^n_c)^T W_\tilde{c} \tilde{s} + q_c(s^n_c) \]

by using the Steihaug-Toiint algorithm, see [26] and [29], the inequality (2.9) and Lemma 2.1 allows to write

\[ \tilde{Q}_c(\tilde{0}) - \tilde{Q}_c(\tilde{s}_c^c) \geq \frac{1}{2}\|W_c^T \nabla q_c(s^n_c)\| \min\left\{ \frac{\|W_c^T \nabla q_c(s^n_c)\|}{\|W_c^T H_c W_c\|}, \delta_c \right\}. \tag{5.11} \]

Accordingly with the notation introduced in Section 4.3 page 44, we have

\[ \tilde{Q}_c(\tilde{s}) = Q_c(W_c \tilde{s}) = q_c(s + s^n_c) = q_c(s) \]

then

\[ \tilde{Q}_c(\tilde{0}) - \tilde{Q}_c(\tilde{s}_c^c) = q_c(s^n_c) - q_c(s_c). \]

By using the standard assumptions and The Cauchy-Schwartz inequality we can write

\[
\|W_c^T H_c W_c\| \leq \|W_c\|^2 \|H_c\| \\
\leq \omega^2 K_3 \\
= \frac{1}{K_7}.
\]
Then (5.11) is
\[
q_c(s^n_c) - q_c(s_c) \geq \frac{1}{2} \| W_e^T \nabla q_c(s^n_c) \| \min \left\{ \frac{\| W_e^T \nabla q_c(s^n_c) \|}{\| W_e^T H_c W_e \|}, \delta_c \right\} \\
\geq \frac{1}{2} \| W_e^T \nabla q_c(s^n_c) \| \min \left\{ \delta_c, K r \| W_e^T \nabla q_c(s^n_c) \| \right\}.
\]
This completes the proof. ∎

At the current iterate \( x_c \), let us denote the actual reduction of \( \ell(x) \) due to the step \( s^t_c = W_c \tilde{s}^t_c \) from \( s^n_c \) as
\[
hAred_c = \ell(x_c + s^n_c) - \ell(x_c + s^n_c + W_c \tilde{s}^t_c)
\]
and the predicted decrease to the quadratic model of \( \ell(x) \) due to the step \( W_c \tilde{s}^t_c \) from \( s^n_c \) by
\[
hPred_c = -\frac{1}{2} (s^n_c)^T W_e^T H_c W_e \tilde{s}^t_c - \nabla q_c(s^n_c)^T W_e \tilde{s}^t_c.
\]
Then (5.10) becomes
\[
hPred_c \geq \frac{1}{2} \| W_e^T \nabla q_c(s^n_c) \| \min \{ \delta_c, K r \| W_e^T \nabla q_c(s^n_c) \| \}. \tag{5.12}
\]

**Lemma 5.6**

Under the standard assumptions, there exists a constant \( K_8 \), which does not depend on \( x_c \) such that
\[
|hAred(s_c) - hPred(s_c)| \leq K_8 \| s_c \|^2. \tag{5.13}
\]

**Proof**

\[
\ell(x_c + s^n_c) - \ell(x_c + s^n_c + W_c \tilde{s}^t_c) = \ell(x_c + s^n_c) - \left[ \ell(x_c + s^n_c) + \nabla \ell(x_c + s^n_c)^T W_c \tilde{s}^t_c + \frac{1}{2} (W_c \tilde{s}^t_c)^T H(x_c + s^n_c + \xi W_c \tilde{s}^t_c) W_c \tilde{s}^t_c \right]
\]
\[
\quad + \frac{1}{2} (W_c \tilde{s}^t_c)^T H(x_c + s^n_c + \xi W_c \tilde{s}^t_c) W_c \tilde{s}^t_c
\]
\[
= -\nabla \ell(x_c + s^n_c)^T W_c \tilde{s}^t_c
\]
\[
- \frac{1}{2} (W_c \tilde{s}^t_c)^T H(x_c + s^n_c + \xi W_c \tilde{s}^t_c) W_c \tilde{s}^t_c,
\]
where \( \xi \in (0, 1) \). Then,
\[
|hAred(s_c) - hPred(s_c)| = |\ell(x_c + s^n_c) - \ell(x_c + s^n_c + W_c \tilde{s}^t_c)|
\]
\[ + \frac{1}{2}(s_c^T W_c^T H_c W_c s_c^T + \nabla q_c(s^n_c)^T W_c s_c^T)] \]

\[ = - \nabla \ell(x_c + s^n_c)^T W_c s_c^T \]

\[ - \frac{1}{2}(W_c^T H(x_c + s^n_c + \xi W_c s_c^T)W_c s_c^T \]

\[ + \frac{1}{2}(s_c^T W_c^T H_c W_c s_c^T + (\nabla \ell_c + H_c s_c^n)^T W_c s_c^T), \]

with \( \eta \in (0, 1) \). Now,

\[ \nabla \ell(x_c + s^n_c) = \nabla \ell_c + H(x_c + \eta s^n_c) s^n_c. \]

Then

\[ |hAred(s_c) - hPred(s_c)| = |\frac{1}{2}(W_c^T H_c - H(x_c + s^n_c + \xi W_c s_c^T)]W_c s_c^T \]

\[ + (s_c^n)^T[\nabla \ell + H(x_c + \eta s^n_c)]W_c s_c^T \]

\[ \leq \frac{1}{2} \|H_c - H(x_c + s^n_c + \xi W_c s_c^T)\| \|W_c\| \|s_c^T\|^2 \]

\[ + \|H_c - H(x_c + \eta s^n_c)\| \|W_c\| \|s_c^n\| \|s_c^T\|. \quad (5.14) \]

Because \( H(x) \) is continuous on a neighborhood \( \mathcal{U}(x_c) \subset \Omega \), there exist constants \( M_1, M_2 > 0 \) such that

\[ \|H_c - H(x_c + s^n_c + \xi s_c)\| \leq M_1 \]

\[ \|H_c - H(x_c + \eta s^n_c)\| \leq M_2. \]

Because \( \|s^n_c\| \leq \|s_c\| \) and \( \|s_c^T\| \leq \|s_c\| \), we have

\[ |hAred(s_c) - hPred(s_c)| \leq \left( \frac{1}{2}M_1 \omega + M_2 \omega \right) \|s_c\|^2 \]

\[ = K \|s_c\|^2. \]

Hence, the result is established. \( \square \)

### 5.3 The decrease in the model

In this section we deal with the decrease of the quadratic model of the merit function. The quantities \( Pred_c \) and \( Ared_c \) have already been defined in Chapter 2.
We observe that if the penalty parameter is bounded the next lemma shows that the predicted reduction provides an approximation to the actual merit function reduction that is accurate to within the square of the steplength. As El-Alem [8], has pointed out the result does not depend on any property of the step.

**Lemma 5.7**

Under the standard assumptions, if \( x_c \) is the current iterate, then

i) 
\[
|\text{Ared}_c(s_c; \rho_c) - \text{Pred}_c(s_c; \rho_c)| \leq L_1 \|s_c\|^2 + L_2 \rho_c \|s_c\|^3 \\
\quad + L_3 \rho_c \|s_c\|^2 \|C_c\| 
\]  
(5.15)

where \( L_1, L_2 \) and \( L_3 \) are constants which do not depend on \( x_c \).

ii) 
\[
|\text{Ared}_c(s_c; \rho_c) - \text{Pred}_c(s_c; \rho_c)| \leq K_0 \rho_c \|s_c\|^2 
\]  
(5.16)

where \( K_0 \) is a constant which does not depend on \( x_c \).

**Proof**

The proofs follow directly from El-Alem [8]. i)

\[
|\text{Ared}_c(s_c; \rho_c) - \text{Pred}_c(s_c; \rho_c)| = \left| \frac{1}{2} \rho_c \left( \|C(x_c)\|^2 - \|C(x_c + C_c)\|^2 \right) \right| 
\]  
(5.17)

Since

\[
h\text{Ared}_c = \ell(x_c + s_c^u) - \ell(x_c + s_c^u + s_c^i) \\
h\text{Pred}_c = q_c(s_c^u) - q_c(s_c),
\]

we have

\[
|\text{Ared}_c(s_c; \rho_c) - \text{Pred}_c(s_c; \rho_c)| = |q_c(s_c^u) - h\text{Pred}_c - \ell(x_c + s_c)|
\]  
(5.18)
\begin{align*}
  &+ \frac{1}{2} \rho_c \left[ \| \nabla C_c^T s_c + C_c \|^2 - \| C(x_c + s_c) \|^2 \right] \\
  &+ \ell(x_c + s^n_c) - \ell(x_c + s^n_c) \\
  = &\ \left[ h_Ared_c - hPred_c + q_c(s^n_c) - \ell(x_c + s^n_c) \right] \\
  &+ \frac{1}{2} \rho_c \left[ \| \nabla C_c^T s_c + C_c \|^2 - \| C(x_c + s_c) \|^2 \right] \\
  \leq &\ \left| h_Ared_c - hPred_c \right| + |q_c(s^n_c) - \ell(x_c + s^n_c)| \\
  &+ \frac{1}{2} \rho_c \left[ \| \nabla C_c^T s_c + C_c \|^2 - \| C(x_c + s_c) \|^2 \right]. \quad (5.17)
\end{align*}

Now,

\begin{align*}
  q_c(s^n_c) - \ell(x_c + s^n_c) &= \frac{1}{2} (s^n_c)^T [ H_c - H(x_c + \eta s^n_c)] s^n_c \\
  |q_c(s^n_c) - \ell(x_c + s^n_c)| &\leq \frac{1}{2} \|[H_c - H(x_c + \eta s^n_c)] \| \| s^n_c \|^2 \\
  &\leq \frac{1}{2} M_2 \| s^n_c \|^2
\end{align*}

where \( \eta \in (0, 1) \) and \( M_2 > 0 \) is as in Lemma 5.6.

\begin{align*}
  \| \nabla C_c^T s_c + C_c \|^2 - \| C(x_c + s_c) \|^2 &= \| \nabla C_c^T s_c \|^2 - \| \nabla C_c^T s_c + C_c \|^2 \\
  &= s_c^T \left[ \sum_{i=1}^{m} C_i(x_c + \xi s_c) \nabla^2 C_i(x_c + \xi s_c) \right] s_c \\
  &= s_c^T \left[ \nabla C_c^T \nabla C_c - \nabla C(x_c + \xi s_c) \nabla C(x_c + \xi s_c)^T \\
  &\quad - \sum_{i=1}^{m} C_i(x_c + \xi s_c) \nabla^2 C_i(x_c + \xi s_c) \right] s_c \\
  \left| \| \nabla C_c^T s_c + C_c \|^2 - \| C(x_c + s_c) \|^2 \right| &\leq \left| \| \nabla C_c^T \nabla C_c - \nabla C(x_c + \xi s_c) \nabla C(x_c + \xi s_c)^T \right| \\
  &\quad + \| \sum_{i=1}^{m} C_i(x_c + \xi s_c) \nabla^2 C_i(x_c + \xi s_c) \| \| s_c \|^2.
\end{align*}

Since the functions \( \nabla C(x) \nabla C(x)^T \) and \( C_i(x) \nabla^2 C_i(x) \) for all \( i = 1, \ldots, m \) are continuous on a neighborhood \( U(x_c) \subset \Omega \) so there exist constants \( M_3 > 0 \) and \( M_4 > 0 \) such that

\begin{align*}
  \| \nabla C_c^T \nabla C_c - \nabla C(x_c + \xi s_c) \nabla C(x_c + \xi s_c)^T \| &\leq M_3 \quad \text{and} \\
  \| \sum_{i=1}^{m} C_i(x_c + \xi s_c) \nabla^2 C_i(x_c + \xi s_c) \| &\leq M_4,
\end{align*}

where \( \xi \in (0, 1) \) and \( x_c \in \Omega \). Then

\begin{align*}
  \left| \| \nabla C_c^T s_c + C_c \|^2 - \| C(x_c + s_c) \|^2 \right| &\leq (M_3 + M_4) \| s_c \|^2.
\end{align*}
Thus (5.17) becomes

\[
|\text{Ared}_e - \text{Pred}_e| \leq |h \text{Ared}_e - h \text{Pred}_e| + \frac{1}{2} \| H_e - H(x_c + \eta s^0_e) \| s^0_e \|^2 \\
+ \frac{1}{2} \rho_c \| \nabla C_e \nabla C^T_e - \nabla C(x_c + \xi s_e) \nabla C(x_c + \xi s_e)^T \| s_e \|^2 \\
+ \frac{1}{2} \rho_c \| \sum_{i=1}^m C_i (x_c + \xi s_e) \nabla^2 C_i (x_c + \xi s_e) \| s_e \|^2 \\
\leq K_8 ||s_e||^2 + \frac{1}{2} [\alpha_2 ||s_e||^2 + \rho_c M_3 ||s_e||^2 + \rho_c M_4 ||s_e||^2].
\]

Since \( ||s^0_e|| \leq ||s_e|| \) and \( \rho_c \geq 1 \), then

\[
|\text{Ared}_e(s_c; \rho_c) - \text{Pred}_e(s_c; \rho_c)| \leq [K_8 + \frac{1}{2}(\alpha_2 + M_3 + M_4)] \rho_c ||s_e||^2 \\
= K_9 \rho_c ||s_e||^2.
\]

This completes the proof. \( \square \)

**Lemma 5.8**

Under the standard assumptions, there exists a constant \( K_{10} \geq 0 \), such that for any \( k \),

\[
q_k(0) - q_k(s^0_k) \geq -K_{10} ||C_k||. \tag{5.18}
\]

**Proof**

Let us compute

\[
q_k(0) - q_k(s^0_k) = -\nabla \ell_k^T s^0_k - \frac{1}{2} (s^0_k)^T H_k s^0_k \\
\geq -||\nabla \ell_k|| ||s^0_k|| - \frac{1}{2} ||H_k|| ||s^0_k||^2 \\
= -(||\nabla \ell_k|| + \frac{1}{2} ||H_k|| ||s^0_k||) ||s^0_k||.
\]

From the fact that \( ||s^0_k|| \leq r \delta_k < \delta_{\text{max}} \), Lemma 5.1 and Lemma 5.3 we have

\[
q_k(0) - q_k(s^0_k) \geq -(K_1 + \frac{1}{2} \delta_{\text{max}} K_3) K_4 ||C_k||.
\]

Defining \( K_{10} = (K_1 + \frac{1}{2} \delta_{\text{max}} K_3) K_4 \) we obtain (5.18).

\[
q_k(0) - q_k(s^0_k) \geq -K_{10} ||C_k||.
\]

\( \square \)
Lemma 5.9

Let \( s_c \) be the step generated by Algorithm 3.4 at the current iterate. Then under the standard assumptions, the decrease in the merit function predicted by the model satisfies

\[
\text{Pred}_c(s_c; r_c) \geq \frac{1}{2} \| W_c^T \nabla q_c(s_c^u) \| \min \{ K_T \| W_c^T \nabla q_c(s_c^u) \| , \delta_c \} \\
- K_{10} \| C_c \| + \frac{r_c}{2} [ \| C_c \|^2 - \| \nabla C_c^T s_c + C_c \|^2 ] .
\]

Proof

Let us recall that

\[
\text{Pred}_c(s_c; r_c) = q_c(0) - q_c(s_c) + \frac{r_c}{2} [ \| C_c \|^2 - \| \nabla C_c^T s_c + C_c \|^2 ] \\
= (q_c(s_c^u) - q_c(s_c)) + (q_c(0) - q_c(s_c^u)) \\
+ \frac{r_c}{2} [ \| C_c \|^2 - \| \nabla C_c^T s_c + C_c \|^2 ] .
\]

From Lemma 5.5 and Lemma 5.8 we have the inequality

\[
\text{Pred}_c(s_c; r_c) \geq \frac{1}{2} \| W_c^T \nabla q_c(s_c^u) \| \min \{ K_T \| W_c^T \nabla q_c(s_c^u) \| , \delta_c \} \\
- K_{10} \| C_c \| + \frac{r_c}{2} [ \| C_c \|^2 - \| \nabla C_c^T s_c + C_c \|^2 ] .
\]

Hence the result is established.

Let us observe that if \( x_c \in \mathbb{F} \), then the predicted reduction does not depend on \( r_c \), so we take \( r_c \) as the penalty parameter from the previous iteration. Now the question that we have to answer is how near to feasibility must an iterate be in order that the penalty parameter need not to be increased. The answer is given by the following lemma.

Lemma 5.10

Let \( x_c \) be the current iterate and let us assume that the algorithm does not terminate at \( x_c \). If \( \| C_c \| \leq \alpha \delta_c \) where \( \alpha \) is a constant that satisfies:

\[
\alpha \leq \min \left\{ \frac{\varepsilon}{3 \delta_{\text{max}}} , \frac{\varepsilon}{3 \omega K_3 K_4 \delta_{\text{max}}} , \frac{\varepsilon}{24 K_{10}} \min \{ \frac{2 K_T \varepsilon}{3 \delta_{\text{max}}} , 1 \} , \frac{\sqrt{3}}{2 K_4} \right\}
\]

where \( \varepsilon \) and \( \delta_{\text{max}} \) are introduced in Algorithm 3.4; \( \varepsilon \) is the tolerance prescribed in the convergence test and \( \delta_{\text{max}} \) is the upper bound of the sequence of trust radii. Then
\[
Pred_c(s_c; \rho_c) \geq \frac{1}{4} \left| \left| W_c^T \nabla q_c(s^n_c) \right| \right| \min \{ K_7 \left| \left| W_c^T \nabla q_c(s^n_c) \right| \right| , \delta_c \} \\
+ \frac{1}{2} \rho_c \left[ \left| \left| C_c \right| \right|^2 - \left| \nabla C_c^T s_c + C_c \right|^2 \right].
\] (5.20)

**Proof**

If the algorithm does not terminate at \( x_c \), then

\[
\left| \left| W_c^T \nabla \ell_c \right| \right| + \left| \left| C_c \right| \right| \geq \varepsilon,
\]

and since \( \left| \left| C_c \right| \right| \leq \alpha \delta_c \) with \( \alpha \leq \frac{\varepsilon}{3 \delta_{\text{max}}} \) we have \( \left| \left| C_c \right| \right| \leq \frac{\varepsilon}{3} \) and the reduced gradient satisfies

\[
\left| \left| W_c^T \nabla \ell_c \right| \right| \geq \frac{2}{3} \varepsilon.
\] (5.21)

Let us compute:

\[
\left| \left| W_c^T \nabla q_c(s^n_c) \right| \right| = \left| \left| W_c^T (\nabla \ell_c + H_c s^n_c) \right| \right| \\
\geq \left| \left| W_c^T \nabla \ell_c \right| \right| - \left| \left| W_c^T H_c s^n_c \right| \right| \\
\geq \frac{2}{3} \varepsilon - \omega K_3 K_4 \left| \left| C_c \right| \right| \\
\geq \frac{2}{3} \varepsilon - \omega K_3 K_4 \alpha \delta_c.
\]

But since

\[ \alpha \leq \frac{\varepsilon}{3 \omega K_3 K_4 \delta_{\text{max}}} \]

it follows that

\[
\left| \left| W_c^T \nabla q_c(s^n_c) \right| \right| \geq \frac{1}{3} \varepsilon.
\] (5.22)

Now, from Lemma 5.8 we have

\[
Pred_c(s_c; \rho_c) \geq \frac{1}{2} \left| \left| W_c^T \nabla q_c(s^n_c) \right| \right| \min \{ \delta_c , K_7 \left| \left| W_c^T \nabla q_c(s^n_c) \right| \right| \} \\
- K_{10} \left| \left| C_c \right| \right| + \frac{1}{2} \rho_c \left[ \left| \left| C_c \right| \right|^2 - \left| \nabla C_c^T s_c + C_c \right|^2 \right] \\
= \frac{1}{4} \left| \left| W_c^T \nabla q_c(s^n_c) \right| \right| \min \{ \delta_c , K_7 \left| \left| W_c^T \nabla q_c(s^n_c) \right| \right| \} \\
+ \frac{1}{4} \left| \left| W_c^T \nabla q_c(s^n_c) \right| \right| \min \{ \delta_c , K_7 \left| \left| W_c^T \nabla q_c(s^n_c) \right| \right| \} \\
- K_{10} \left| \left| C_c \right| \right| + \frac{1}{2} \rho_c \left[ \left| \left| C_c \right| \right|^2 - \left| \nabla C_c^T s_c + C_c \right|^2 \right].
\]
Using the inequality (5.22) in the second term, we obtain
\[
\text{Pred}_c(s_c; \rho_c) \geq \frac{1}{4} \|W_c^T \nabla q_c(s_c^\alpha)\| \min \{\delta_c^* \cdot K_7 \|W_c^T \nabla q(s_c^\alpha)\| \}
+ \frac{1}{2} \mu \min \left\{ \alpha \cdot \frac{\varepsilon K_7}{3} \right\} \\
- K_{10} \alpha \delta_c + \frac{1}{2} \rho_c [\|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2].
\]

Now because \( \alpha \leq \frac{\sqrt{3}}{2K_4} \delta_c \), we have
\[
\delta_c^2 = \delta_c^2 - \|s_c^\alpha\|^2 \\
\geq \delta_c^2 - K_4^4 \|C_c\|^2 \\
\geq \delta_c^2 - \left( \frac{\sqrt{3}}{2K_4} \right)^2 \alpha^2 \delta_c^2 \\
= \frac{1}{4} \delta_c^2.
\]

Thus
\[
\text{Pred}_c(s_c; \rho_c) \geq \frac{1}{4} \|W_c^T \nabla q_c(s_c^\alpha)\| \min \{\delta_c^* \cdot K_7 \|W_c^T \nabla q(s_c^\alpha)\| \}
+ \frac{\varepsilon \delta_c}{24} \min \left\{ 1, \frac{2 \varepsilon K_7}{3 \delta_{\max}} \right\} \\
- K_{10} \alpha \delta_c + \frac{1}{2} \rho_c [\|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2],
\]

and since
\[
\alpha \leq \frac{\varepsilon}{24K_{10}} \min \left\{ \frac{2K_7 \varepsilon}{3 \delta_{\max}}, 1 \right\},
\]
we have (5.20)
\[
\text{Pred}_c(s_c; \rho_c) \geq \frac{1}{4} \|W_c^T \nabla q_c(s_c^\alpha)\| \min \{K_7 \|W_c^T \nabla q(s_c^\alpha)\|, \delta_c^* \}
+ \frac{1}{2} \rho_c [\|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2].
\]

\[\square\]

**Remark:**

The result given by Lemma 5.10 guarantees that if \( \|C_c\| \leq \alpha \delta_c \) but \( \|W_c^T \nabla \ell_c\| + \|C_c\| \geq \varepsilon \), then the penalty parameter at the current iteration \( x_c \) does not need to be increased in step 2. of Algorithm 3.4.
Lemma 5.11

There exists a constant $K_{11}$, such that for any $k$ for which the algorithm does not terminate and $\|C_k\| \leq \alpha \delta_k$ where $\alpha$ is as in Lemma 5.10, the following inequality holds

$$\text{Pred}_k(s; p) \geq K_{11} \delta_k.$$  \hfill (5.23)

Proof

Since the algorithm does not terminate and $\|C_k\| \leq \alpha \delta_k$, (5.20) becomes

$$\text{Pred}_k(s_k; p_k) \geq \frac{1}{12} \varepsilon \min \left\{ \delta_k, \frac{K_{11} \varepsilon}{3} \right\}$$

$$= \frac{1}{12} \varepsilon \min \left\{ \frac{1}{2} \delta_k, \frac{K_{11} \varepsilon}{3} \right\}$$

$$= \frac{1}{12} \varepsilon \min \left\{ \frac{1}{2}, \frac{K_{11} \varepsilon}{3 \delta_k} \right\} \delta_k$$

$$\geq \frac{1}{12} \varepsilon \min \left\{ \frac{1}{2}, \frac{K_{11} \varepsilon}{3 \delta_{\text{max}}} \right\} \delta_k.$$ \hfill (5.24)

Defining

$$K_{11} = \frac{1}{12} \varepsilon \min \left\{ \frac{1}{2}, \frac{K_{11} \varepsilon}{3 \delta_{\text{max}}} \right\},$$

we have (5.23)

$$\text{Pred}_k(s; p) \geq K_{11} \delta_k.$$  \hfill \square

Summarizing, we have proved that at each iteration the algorithm generates a step $s = s^n + s^t$, such that it gives at least a fraction of Cauchy decrease on the quadratic model of the linearized constraint and that the tangential component $s^t$ predicts, from $s^n$ at least a fraction of Cauchy decrease in the quadratic model of the Lagrangian function on $\mathcal{N}(\nabla C^T)$. We have also proved that if $\|C_k\| \leq \alpha \delta_k$, where $\alpha$ is a small constant which does not depend on the iterates, then the penalty parameter does not need to be increased and a lower bound for the amount of predicted decrease can be found. Let us conclude this chapter by making an observation about the role of the constant $\alpha$ in the inequality $\|C_k\| \leq \alpha \delta_k$. The constant $\alpha$, which depends on the tolerance $\varepsilon$, gives a relative proportion between optimality and feasibility. If the
algorithm does not terminate at $x_c$ and $\|C_c\| \leq \alpha \delta_c$ holds, from the definition of $\alpha$
we obtain (5.21) and then
\[
\frac{\|C_c\|}{\|W_c^T \nabla \ell_c\|} \leq \frac{1}{2}.
\]
In the next chapter we will discuss the role of the penalty parameter in the global
convergence of the nonlinear programming
Chapter 6

The behavior of the penalty parameter

In this chapter we discuss the behavior of the penalty parameter. Here the crucial results is that the sequence of trust region radii \( \{ \delta_k \} \) is bounded away from zero provided that the penalty parameter is increased. This will allow us to conclude that the nondecreasing sequence \( \{ \rho_k \} \) is increased finitely many times.

According to the rule for updating the penalty parameter that we have presented in Section 3.7, we keep the penalty parameter from the previous iteration if the amount of predicted decrease in the quadratic model of the merit function, with the old penalty parameter is at least a fraction of Cauchy decrease in the quadratic model of the linearized constraints, that is, if

\[
Pred_c(s_c; \rho_-) \geq \frac{\rho_-}{4} [||C_c||^2 - ||\Theta_c||^2] \quad \text{then} \quad \rho_c = \rho_-.
\]

The next lemma shows that the converse is also true.

**Lemma 6.1**

Under the standard assumptions, if the penalty parameter is not increased, i.e., if

\[
\rho_- \geq \tilde{\rho}_c
\]

then

\[
Pred_c(s_c; \rho_-) \geq \frac{\rho_-}{4} [||C_c||^2 - ||\Theta_c||^2]. \quad (6.1)
\]

**Proof**

Since the penalty parameter is not increased

\[
\rho_c = \rho_- \geq \frac{4[ q_c(s_c) - q_c(0) ]}{||C_c||^2 - ||\Theta_c||^2}.
\]

Then

\[
\frac{\rho_-}{4} [||C_c||^2 - ||\Theta||^2] + q_c(0) - q_c(s_c) \geq 0
\]

\[
q_c(0) - q_c(s_c) + \frac{\rho_-}{2} [||C_c||^2 - ||\Theta_c||^2] \geq \frac{\rho_-}{4} [||C_c||^2 - ||\Theta_c||^2],
\]

\]
and we obtain
\[ \text{Pred}_c(s_c; \rho_-) \geq \frac{\rho_-}{4} [\|C_c\|^2 - \|\Theta_c\|^2]. \]

This completes the proof. \(\square\)

**Lemma 6.2**

Under the standard assumptions, if \(\text{Pred}_c(s_c; \rho_-) < \frac{\rho_-}{4} [\|C_c\|^2 - \|\Theta_c\|^2]\) then
\[ \text{Pred}_c(s_c; \rho_c) \geq \frac{\rho_c}{4} [\|C_c\|^2 - \|\Theta_c\|^2] \]

where \(\rho_c = \tilde{\rho}_c + \beta\).

**Proof**

If \(\text{Pred}_c(s_c; \rho_-) < \frac{\rho_-}{4} [\|C_c\|^2 - \|\Theta_c\|^2]\), then the penalty parameter is increased:
\[ \rho_c = \frac{4}{\|C_c\|^2 - \|\Theta_c\|^2} [q_c(s_c) - q_c(0)] + \beta. \]

Thus,
\[ \frac{\rho_c}{4} [\|C_c\|^2 - \|\Theta\|^2] + q_c(0) - q_c(s_c) = \frac{\rho_-}{4} [\|C_c\|^2 - \|\Theta\|^2] \]
\[ q_c(0) - q_c(s_c) + \frac{\rho_c}{2} [\|C_c\|^2 - \|\Theta_c\|^2] = \frac{\rho_c}{4} [\|C_c\|^2 - \|\Theta_c\|^2] + \frac{\beta}{4} [\|C_c\|^2 - \|\Theta_c\|^2]. \]

Then it is clear that
\[ \text{Pred}_c(s_c; \rho_c) \geq \frac{\rho_c}{4} [\|C_c\|^2 - \|\Theta_c\|^2] \geq \frac{\rho_-}{4} [\|C_c\|^2 - \|\Theta_c\|^2]. \] (6.2)

Hence the result is established. \(\square\)

The following lemma shows that if the penalty parameter is increased, it is increased by a quantity at least as large as \(\beta\).

**Lemma 6.3**

Under the standard assumptions, if \(\text{Pred}_c(s_c; \rho_-) < \frac{\rho_-}{4} [\|C_c\|^2 - \|\Theta_c\|^2]\) then \(\rho_c - \rho_- \geq \beta\).
Proof
From Lemma 6.1, observe that if $P_{red}(s_c; \rho_-) < \frac{\rho_-}{4} \|[\Theta_c]^2 - \|\Theta_c\|^2\|$, then $\rho_- < \bar{\rho} + \beta$, so the penalty parameter is increased,

$$P_{red}(s_c; \rho_-) \leq \frac{\rho_-}{4} \|[\Theta_c]^2 - \|\Theta_c\|^2\|$$

$$q_c(0) - q_c(s_c) + \frac{\rho_-}{2} \|[\Theta_c]^2 - \|\Theta_c\|^2\| \leq \frac{\rho_-}{4} \|[\Theta_c]^2 - \|\Theta_c\|^2\|$$

$$4[q_c(0) - q_c(s_c)] + 2\rho_- \|[\Theta_c]^2 - \|\Theta_c\|^2\| \leq \rho_- \|[\Theta_c]^2 - \|\Theta_c\|^2\|$$

$$\frac{4[q_c(0) - q_c(s_c)]}{\|[\Theta_c]^2 - \|\Theta_c\|^2\|} \leq -\rho_-.$$

Then,

$$\rho_c \geq \rho_-$$

$$\rho_c - \rho_- \geq 0$$

$$\rho_c - \rho_- \geq \beta.$$  \hspace{1cm} (6.3)

That is, from the previous iteration to the current one the penalty parameter is increased by a quantity at least as large as $\beta$.

\[ \square \]

Lemma 6.4
Let $x_c$ be the current iterate. If $\rho_c$ is increased at $x_c$, then there exists a constant $K_{12}$ which does not depend on $x_c$, such that

$$\rho_c \delta_c \leq K_{12}.$$  \hspace{1cm} (6.4)

Proof
Since

$$\rho_c = \frac{4[q_c(s_c) - q_c(0)]}{\|[\Theta_c]^2 - \|\Theta_c\|^2\|} + \beta,$$

$$\|[\Theta_c]^2 - \|\Theta_c\|^2\| \frac{\rho_c}{4} = [q_c(s_c) - q_c(0)] + \frac{\beta}{4} \|[\Theta_c]^2 - \|\Theta_c\|^2\|$$

$$\|[\Theta_c]^2 - \|\Theta_c\|^2\| \frac{\rho_c}{4} = [q_c(s_c) - q_c(s_c)] + [q_c(s_c) - q(0)]$$

$$+ \frac{\beta}{4} [-2(\nabla C\mathcal{C}T s_c - \|\nabla C\mathcal{C}T s_c\|^2].$$
Applying Corollary 4.1 to the left-hand side, and Lemma 5.5 and Lemma 5.8 to the right-hand side, we can obtain the following:

\[
\frac{K_5}{4} \|C_c\| \min \{r \delta_c, K_\alpha \|C_c\|\} \rho_c \leq \frac{1}{2} \|W_c^T \nabla q_c(s^*_c)\| \min \left\{ \frac{\|W_c^T \nabla q_c(s^*_c)\|}{K_7}, \delta_c \right\} \\
+ K_{10} \|C_c\| - \frac{\beta}{2} (\nabla C_c C_c)^T s_c - \frac{\beta}{4} \|\nabla C_c^T s_c\|^2 \\
\leq K_{10} \|C_c\| - \frac{\beta}{2} (\nabla C_c C_c)^T s_c \\
\leq K_{10} \|C_c\| + \frac{\beta}{2} \|\nabla C_c\| \|C_c\| \|s_c\| \\
\leq (K_{10} + \frac{\beta}{2} \|\nabla C_c\| \|s_c\|) \|C_c\|.
\]

Then,

\[
\frac{K_5}{4} \min \{r \delta_c, \ K_\alpha \|C_c\|\} \rho_c \leq K_{10} + \frac{\beta}{2} \nu_3 \|s_c\| \leq K_{10} + \frac{\beta}{2} \nu_3 \delta_{\text{max}}.
\]

Since at the current iterate the penalty parameter increases, from Lemma 5.10 we must have \(\|C_c\| > \alpha \delta_c\), and so

\[
\frac{K_5}{4} \min \{r \delta_c, \ K_\alpha \alpha \delta_c\} \rho_c \leq K_{10} + \frac{\beta}{2} \nu_3 \delta_{\text{max}}
\]

\[
\rho_c \delta_c \leq \frac{4K_{10} + 2\beta \nu_3 \delta_{\text{max}}}{K_5 \min \{r, K_\alpha \alpha\}}.
\]

Defining

\[
K_{12} = \frac{4K_{10} + 2\beta \nu_3 \delta_{\text{max}}}{K_5 \min \{r, K_\alpha \alpha\}}
\]

we obtain (6.4).

The following lemma gives a lower bound for the sequence \(\{\delta_k\}\) provided that the algorithm does not terminate and the penalty parameter is increased.

**Lemma 6.5**

Under the standard assumptions, there exists a constant \(\delta\), which does not depend on the iterates, such that for any \(k\) at which the algorithm does not terminate and the penalty parameter is increased,

\[
\delta_k \geq \delta.
\] (6.5)
Proof
Let $s_{k-1}$ be the last acceptable step, $(x_k = x_{k-1} + s_{k-1})$, and we are trying to find a new acceptable step $s_k$. In the process of finding $s_k$ we have generated a set of unacceptables trial steps. Let us denote the indices of unacceptables steps by $k_1, k_2, \ldots, k_j$. That is

$$s_{k-1}, s_{k_1}, s_{k_2}, \ldots, s_{k_j}, s_{k_{j+1}} = s_k.$$  

We must consider three different situations:

i) $s_{k_1} = s_k$, that is, there is no unacceptable step.

ii) $s_{k_1} \neq s_k$ and $\|C_k\| > \alpha \delta_k$ for all $i = 1, \ldots, j + 1$.

iii) $s_{k_1} \neq s_k$ but $\|C_k\| > \alpha \delta_k$ does not hold for all $i = 1, \ldots, j + 1$.

i) If $s_k = s_{k_1}$, then from the way of updating the trust region radius,

$$\delta_k \geq \max\{\alpha \|s_{k-1}\|, \delta_{\text{min}}\} \geq \delta_{\text{min}}.$$  

(6.6)

ii) If $s_{k_1} \neq s_k$ and at the same time the constraint violation $\|C_k\| > \alpha \delta_k$ for all $i = 1, \ldots, j$ then from Lemma 5.7 we have

$$|\text{Aread}_k(s_{k_1}; \rho_k) - \text{Pred}_k(s_{k_1}; \rho_k)| \leq K_0 \rho_k \|s_{k_1}\|^2.$$  

Now since $\|C_k\| > \alpha \delta_k$, from (6.2) and Corollary 5.1 we have

$$\text{Pred}_k(s_{k_1}; \rho_k) \geq \frac{\rho_k}{4} [\|C_k\|^2 - \|\Theta_k\|^2]$$

$$\geq \frac{\rho_k}{4} K^3 \|C_k\| \min\{K_0 \alpha, \rho \delta_k\}$$

$$= \frac{\rho_k}{4} K^3 \|C_k\| \min\{K_0 \alpha, \rho \delta_k\}.$$  

Then

$$\frac{|\text{Aread}_k(s_{k_1}; \rho_k) - \text{Pred}_k(s_{k_1}; \rho_k)|}{\text{Pred}_k(s_{k_1}; \rho_k)} \leq \frac{4K_0 \rho_k \delta_k \|s_{k_1}\|}{\rho_k \|C_k\| \min\{K_0 \alpha, \rho \delta_k\} \delta_k}$$

$$\leq \frac{4K_0 \|s_{k_1}\|}{K^3 \|C_k\| \min\{K_0 \alpha, \rho \delta_k\}}.$$  

(6.7)

Since all the steps $s_k$ for $i = 1, \ldots, j$ are rejected, it must be the case that

$$1 - \eta_1 < \frac{|\text{Aread}_k(s_{k_1}; \rho_k)|}{\text{Pred}_k(s_{k_1}; \rho_k)} - 1,$$  

(6.8)
so from (6.7) and (6.8) we have that
\[
\|s_k\| \geq \frac{(1 - \eta_1)K_5 \min\{\alpha K_6, r\}}{4K_9}\|C_k\|, \quad \forall i = 1, \ldots, j. \tag{6.9}
\]

Since \(\delta_{k+1} = \alpha_1\|s_k\|\), and since \(\|C_k\| > \alpha \delta_{k_1}\), it follows that
\[
\delta_k = \alpha_1\|s_k\| \geq \alpha_1 \left[\frac{(1 - \eta_1)K_5 \min\{\alpha K_6, r\}}{4K_9}\right] \alpha \delta_{k_1}. \tag{6.10}
\]

Now, according to the rule for updating the trust region radius,
\[
\delta_{k_1} = \max\{\alpha_1\|s_{k-1}\|, \delta_{\min}\} \geq \delta_{\min}.
\]

Then in (6.10)
\[
\delta_k \geq \frac{\alpha_1(1 - \eta_1)K_5 \min\{\alpha K_6, r\}}{4K_9} \alpha \delta_{\min}
\]
\[
= \ K_{13}. \tag{6.11}
\]

iii) If \(s_{k_1} \neq s_k\) and \(\|C_k\| > \alpha \delta_{k_1}\), does not hold for all \(i = 1, \ldots, j + 1\), then there exists at least one index \(p\) such that \(\|C'_p\| \leq \alpha \delta_{k_p}\).

Let \(l\) be the largest index such that \(\|C'_l\| \leq \alpha \delta_{k_l}\).

Let \(l\) be the largest index such that \(\|C'_l\| \leq \alpha \delta_{k_l}\) holds.

\[
\text{rejected}
\]
\[
\text{rejected}
\]
\[
\|C_k\| \geq \alpha \delta_{k_i}, \quad \|C_k\| > \alpha \delta_{k_i}
\]

Observe that for all indices \(i\) such that \(l + 1 \leq i \leq j + 1\), we have \(\|C_k\| > \alpha \delta_{k_i}\), then as in the first two cases we will see that there exists a constant \(K_{14}\) such that
\[
\delta_k \geq K_{14}\|s_k\|, \tag{6.12}
\]

If \(s_{k+1} = s_k\), then
\[
\delta_k \geq \max\{\alpha_1\|s_k\|, \delta_{\min}\}
\]
\[
\geq \alpha_1\|s_k\|. \tag{6.13}
\]

If \(s_{k+1} \neq s_k\), since \(\|C_k\| > \alpha \delta_{k_i}\) for all \(i = l + 1, \ldots, j + 1\) from (6.10), we have
\[
\delta_k \geq \alpha_1 \left[\frac{(1 - \eta_1)K_5 \min\{\alpha K_6, r\}}{4K_9}\right] \alpha \delta_{k_i}
\]
\[
\geq \alpha_1 \left[\frac{(1 - \eta_1)K_5 \min\{\alpha K_6, r\}}{4K_9}\right] \alpha\|s_k\|. \tag{6.14}
\]
Then from (6.13) and (6.14), defining

$$K_{14} = \min \left\{ \alpha_1, \alpha_1 \left[ \frac{(1 - \eta_1)K_5 \min \{\alpha \cdot K_6, r\}}{4K_9} \right] \alpha \right\}$$

we obtain (6.12).

Now, since $\|C_k\| \leq \alpha \delta_k$, the penalty parameter does not need to be increased.

From (5.15) we have

$$|\text{Red}_k(s_k; \rho_k) - \text{Pred}_k(s_k; \rho_k)|$$

$$\leq L_1 \|s_k\|^2 + L_2 \rho_k \|s_k\|^2 + L_3 \rho_k \|s_k\|^2 \|C_k\|$$

$$\leq L_1 \|s_k\|^2 + L_2 \rho_k \|s_k\| \|s_k\|^2 + L_3 \rho_k \|s_k\|^2 \alpha \delta_k$$

$$\leq [L_1 \|s_k\|^2 + (L_2 + \alpha L_3) \rho_k \|s_k\|^2 \delta_k]. \quad (6.15)$$

Observe that

$$\rho_k \|s_k\| \leq \rho_k \frac{\delta_k}{K_{14}} \leq \frac{K_{12}}{K_{14}} = L_4$$

because at iteration $k$ the penalty parameter is increased by hypothesis. Then (6.15) becomes

$$|\text{Red}_k(s_k; \rho_k) - \text{Pred}_k(s_k; \rho_k)|$$

$$\leq [L_1 + L_2 L_4 + \alpha L_3 L_4] \|s_k\| \delta_k. \quad (6.16)$$

Also, since $\|C_k\| \leq \alpha \delta_k$, from Lemma 5.11 we know that

$$\text{Pred}_k(s_k; \rho_k) \geq K_{11} \delta_k. \quad (6.17)$$

Then from (6.16) and (6.17) and from the fact that $s_k$ is rejected we obtain

$$(1 - \eta_1) < \left| \frac{\text{Red}_k(s_k; \rho_k)}{\text{Pred}_k(s_k; \rho_k)} - 1 \right|$$

$$\leq \frac{|\text{Red}_k(s_k; \rho_k) - \text{Pred}_k(s_k; \rho_k)|}{\text{Pred}_k(s_k; \rho_k)}$$

$$\leq \frac{[L_1 + L_2 L_4 + L_3 \alpha L_4] \|s_k\|^2}{K_{11} \delta_k}$$

$$\leq \frac{[L_1 + L_2 L_4 + L_3 \alpha L_4] \|s_k\|}{K_{11}}.$$
Then
\[ \|s_k\| \geq \frac{(1 - \eta_1)K_{11}}{L_1 + L_2 L_4 + L_3 \alpha L_4}. \] (6.18)

Now using (6.12) and (6.18) we obtain the following:
\[ \delta_k \geq K_{14}\|s_k\| \]
\[ \geq \frac{(1 - \eta_1)K_{11} K_{14}}{L_1 + L_4 L_4 + L_3 \alpha L_4} \]
\[ = K_{15}. \]

Defining
\[ \hat{\delta} = \min\{\delta_{\min}, K_{13}, K_{15}\} \]
we obtain the desired result. \( \square \)

Lemma 6.4 and Lemma 6.5 will allow us to show that the nondecreasing sequence of penalty parameters generated by the nonlinear programming Algorithm 3.4 is bounded.

**Lemma 6.6**

Under the standard assumptions, if the algorithm does not terminate then
\[ \lim_{k \to \infty} \rho_k = \rho^* < \infty. \] (6.19)

**Proof**

If the penalty parameter at \( x_k \) is increased, then from Lemma 6.4 and Lemma 6.5 we have
\[ \rho_k \delta_k \leq K_{12} \quad \text{and} \quad \delta_k \geq \hat{\delta} \]
\[ \rho_k \leq \frac{K_{12}}{\delta_k} \leq \frac{K_{12}}{\hat{\delta}}. \]

Therefore \( \{\rho_k\} \) is a bounded sequence and since that it is nondecreasing there exists \( \rho^* < \infty \) such that
\[ \lim_{k \to \infty} \rho_k = \rho^*. \]

This completes the proof. \( \square \)
Corollary 6.1

Under the standard assumptions, if the algorithm does not terminate, then the penalty parameter is increased only at a finite number of iterations.

Proof

Accordingly to the scheme for updating the penalty parameter and Lemma 6.1, we know that \( \rho_0 > \bar{\rho}_c + \beta \) if and only if \( \text{Pred}_c(s_c; \rho_0) \geq \frac{\lambda_\infty}{4} [||C_c||^2 - ||\Theta_c||^2] \). From Lemma 6.3 we know that if the penalty parameter is increased, it is increased by a quantity greater than or equal to \( \beta \), (that is \( \rho_k - \rho_{k-1} \geq \beta \)) and since the sequence \( \{\rho_k\} \) converges to \( \rho^* < \infty \) the number of iterations at which the penalty parameter is increased must be finite. Thus there exists an index \( k_\rho \) such that

\[
\rho_k = \rho_{k_\rho}, \quad \forall k \geq k_\rho, \tag{6.20}
\]

and we obtain the desired result. \( \square \)

This last result will play a crucial role in the proof of the global convergence of Algorithm 3.4.
Chapter 7

The global convergence theorems

In this chapter we begin by showing that the general nonlinear programming algorithm is well-defined. In Section 7.2 we will present more properties of the trust region radius sequence generated by the algorithm. In Section 7.3 we establish the global convergence theorem.

7.1 The finite termination theorem

The next theorem shows that the nonlinear programming Algorithm 3.4 is well-defined in the sense that at each iteration we can find an acceptable step after a finite number of iterations.

Theorem 7.1

Under the standard assumptions, unless some iterate \(x_k\) satisfies the termination condition of the Algorithm 3.4, an acceptable step from \(x_k\) will be found after finitely many trust region subproblem solutions.

Proof

The proof follows from Theorem 5.1 of El-Alem [9]. Let \(x_c\) the current iteration and suppose that the algorithm does not terminate at \(x_c\). We will consider two cases accordingly with \(\|C_c\| > \alpha \delta_c\) or \(\|C_c\| \leq \alpha \delta_c\), where \(\alpha\) is as in Lemma 5.10.

Case I: If \(\|C_c\| > \alpha \delta_c\), from (6.2) and Corollary 5.1 we can write

\[
\begin{align*}
\text{Pred}_c & \geq \frac{\rho_c}{4} \|C_c\|^2 - \|\Theta_a\|^2 \\
& \geq \frac{\rho_c}{4} K_h \|C_c\| \min\{K_h \alpha \delta_c, \, r \delta_c\} \\
& = \frac{\rho_c}{4} K_h \|C_c\| \min\{K_h \alpha, \, r\} \delta_c.
\end{align*}
\]

From the last inequality and Lemma 5.7,

\[
\frac{|Ax_{c} - \text{Pred}_c|}{\text{Pred}_c} \leq \frac{4K_h \rho_c \delta_c^2}{\rho_c K_h \|C_c\| \min\{K_h \alpha, \, r\} \delta_c}
\]
\[
\leq \frac{4K_9\delta_c}{K_5\|C_c\| \min\{K_6\alpha_v, r\}}. \tag{7.1}
\]

**Case II:** If \(\|C_c\| \leq \alpha \delta_c\), from Lemma 5.11 we know that \(\text{Pred}_c \geq K_{11}\delta_c\), then
\[
\frac{|\text{Armed}_c - \text{Pred}_c|}{\text{Pred}_c} \leq \frac{K_9\rho_c \delta_c^2}{K_{11}\delta_c} \leq \frac{K_9\rho_c}{K_{11}} \delta_c. \tag{7.2}
\]

Now, in both cases (7.1) and (7.2), as \(\delta_c\) becomes smaller, the quantity
\[
\frac{|\text{Armed}_c - \text{Pred}_c|}{\text{Pred}_c} = \left|\frac{\text{Armed}_c}{\text{Pred}_c} - 1\right|
\]
goess to zero and so the condition \(\frac{\text{Armed}_c}{\text{Pred}_c} \geq \eta_1\) will be met after a finite number of iterations. \(\Box\)

### 7.2 Properties of the sequence \(\delta_k\)

**Lemma 7.1**

Under the standard assumptions, if \(\{\delta_k\}\) converges to zero then both sequences \(\{\|s^n_k\|\}\) and \(\{\|s^l_k\|\}\) converge to zero.

**Proof**

The proof is straightforward.

Since \(\|s^n_k\| \leq r\delta_k\), and \(\|s^l_k\| \leq \delta_k = \sqrt{\delta^2 - \|s^n_k\|^2}\), if \(\delta_k\) converges to zero, so do \(\|s^n_k\|\) and \(\|s^l_k\|\). \(\Box\)

**Lemma 7.2**

Under the standard assumptions, if the sequence \(\{\|C_k\|\}\) is bounded away from zero, then \(\{\delta_k\}\) is also bounded away from zero.

**Proof**

The proof is by contradiction. Let us suppose that
\[
\liminf_{k \to \infty} \delta_k = 0. \tag{7.3}
\]
This limit and the fact that the trust region radius decreases only when the step is rejected, imply that there exists a subsequence of unacceptable steps that satisfies
\[ \delta_{k_j} < \sigma (1 - \eta_1), \] (7.4)
where \( \sigma > 0 \) is a constant which does not depend on \( k \).

Since \( \| C_k \| \) is bounded away from zero, then there exist \( \varepsilon_1 > 0 \) and \( N_1 > 0 \) such that \( \| C_k \| \geq \varepsilon_1 \) for all \( k \geq N_1 \).

Since \( \{ \delta_{k_j} \} \) does converge to zero, \( \text{Pred}_{k_j} > 0 \), and the steps \( s_{k_j} \) are not accepted, we have
\[ \frac{\text{Ared}_{k_j}}{\text{Pred}_{k_j}} < \eta_1. \]

Thus
\[ 1 - \eta_1 < 1 - \frac{\text{Ared}_{k_j}}{\text{Pred}_{k_j}} = \frac{|\text{Ared}_{k_j} - \text{Pred}_{k_j}|}{\text{Pred}_{k_j}} \leq \frac{K_0 \rho_k \| s_{k_j} \|^2}{\text{Pred}_{k_j}}, \] (7.5)
The last inequality comes from Lemma 5.7.

We are going to find a lower bound for \( \text{Pred}_{k_j} \) of the form
\[ \text{Pred}_{k_j} \geq \sigma \rho_k \delta_{k_j}, \]
where \( \sigma > 0 \) as in (7.4). Then, using this bound and the relation given by (7.5), we will be able to obtain a contradiction of the hypothesis (7.3).

Inequality (6.2), Corollary 4.1 and the fact that \( \| C_k \| \) is bounded away from zero, that is, \( \| C_k \| \geq \varepsilon_1 \), allow us to write
\[ \text{Pred}_{k_j} \geq \frac{\rho_k}{4} \left[ \| C_k \|^2 - \| \Theta_k \|^2 \right] \]
\[ \geq \frac{\rho_k}{4} K_5 \| C_k \| \min \left\{ K_6 \| C_k \|, \frac{\varepsilon_1}{\delta_{k_j}} \right\} \delta_{k_j} \]
\[ \geq \frac{\rho_k}{4} K_5 \varepsilon_1 \min \left\{ \frac{K_6}{\delta_{k_j}}, \frac{\varepsilon_1}{\delta_{k_j}} \right\} \delta_{k_j} \]
\[ \geq \frac{\rho_k}{4} K_5 \varepsilon_1 \min \left\{ \frac{K_6}{\delta_{\max}}, \frac{\varepsilon_1}{\delta_{\max}} \right\} \delta_{k_j} \]
\[ = \sigma_4 \rho_k \delta_{k_j}. \] (7.6)
where
\[
\sigma_2 = \frac{K_2 \varepsilon_1}{4} \min \left\{ \frac{K_6}{3\delta_{\text{max}}}, r \right\}.
\]

Then, in (7.5), for all \(k_j \geq N_1\),
\[
1 - \eta_1 < \frac{K_9 \rho_{k_j} \|s_{k_j}\|^2}{P_{\text{red}_{k_j}}}
\leq \frac{K_9 \rho_{k_j} \|s_{k_j}\|^2}{\sigma_2 \rho_{k_j} \delta_{k_j}}
\leq \frac{K_9 \delta_{k_j}}{\sigma_2}.
\]

Now if we take \(\sigma\) in (7.4) to be equal to \(\frac{\alpha \rho_{k_j}}{\delta_{k_j}}\), we obtain a contradiction. Hence the Lemma is proved.

In Lemma 6.5, we have proved that the sequence \(\{\delta_k\}\) is bounded away from zero provided that \(\rho_k\) is increased. This result allowed us to show that the nondecreasing sequence \(\{\rho_k\}\) is bounded above.

The following lemma shows that the sequence \(\{\delta_k\}\) is bounded away from zero not only for the iterates at which the penalty parameter is increased. The main tool in its proof is precisely the fact that we already know that the sequence \(\{\rho_k\}\) is bounded.

**Lemma 7.3** Under the standard assumptions let us assume that at the current iterate the algorithm does not terminate. Then there exists a constant \(\delta_*\) which does not depend on the iterates such that
\[
\delta_k \geq \delta_* \quad \forall k.
\]  

**Proof**
The proof follows by using the same technique as in Lemma 6.5. We have three cases to consider.

i) \(s_{k_i} = s_k\), that is, there is no unacceptable step in between; then accordingly to the rule for updating the trust region radius we have
\[
\delta_k \geq \max\{\alpha_1\|s_{k-1}\|, \delta_{\text{min}}\} \geq \delta_{\text{min}}.
\]  

ii) If \(s_{k_i} \neq s_k\) and \(\|C_k\| > \alpha \delta_{k_i}\) for all \(i = 1, \ldots, j\), the proof is exactly the same as before; see ii) in Lemma 6.5, page 72, for details.
iii) Now, if \( s_k \neq s_k \) and \( \|C_k\| > \alpha \delta_k \) does not hold for all \( i = 1, \ldots, j \), as in Lemma 6.5, let \( l \) be the largest index such that \( \|C_k\| \leq \alpha \delta_k \) holds, and we obtain
\[
\delta_k \geq K_{14} \|s_k\|, \tag{7.9}
\]
where \( K_{14} \) is a constant which does not depend on \( k \).

Now
\[
|\text{Ar}e\text{d}_k(s_k; \rho_k) - \text{Pr}e\text{d}_k(s_k; \rho_k)| \leq K_9 \rho_k \|s_k\|^2
\]
and since \( \|C_k\| \leq \alpha \delta_k \), the penalty parameter does not need to be increased and then \( \text{Pr}e\text{d}_k(s_k; \rho_k) \geq K_{11} \delta_k \).

The step \( s_k \) is an unacceptable step, so
\[
(1 - \eta_1) \leq \frac{|\text{Ar}e\text{d}_k(s_k; \rho_k) - \text{Pr}e\text{d}_k(s_k; \rho_k)|}{\text{Pr}e\text{d}_k(s_k; \rho_k)} \leq \frac{K_9 \rho_k \|s_k\|^2}{K_{11} \delta_k}
\]
\[
\leq \frac{K_9 \rho^* \|s_k\|}{K_{11}}. \tag{7.10}
\]

The last inequality comes from the fact that we already know that the penalty parameter is bounded. The relation (7.10) implies that
\[
\|s_k\| \geq \frac{(1 - \eta_1)K_{11}}{K_9 \rho^*}.
\]

This last inequality and (7.9) allow us to write
\[
\delta_k \geq \frac{(1 - \eta_1)K_{11}K_{14}}{K_9 \rho^*}. \tag{7.11}
\]

From (6.5), (7.8) and (7.11) we define
\[
\delta_\ast = \min \left\{ \hat{\delta}, \delta_{\text{min}}, \frac{(1 - \eta_1)K_{11}K_{14}}{K_9 \rho^*} \right\}.
\]
We have \( \delta_k \geq \delta_\ast \), which is the desired result.
Lemma 7.4
Under the standard assumptions, if the algorithm does not terminate and
the sequence \( \{\|C_k\|\} \) converges to zero then \( \|s_k\| \) is bounded away from
zero.

Proof
Since \( \|C_k\| \) goes to zero as \( k \) goes to infinity, there exists an index \( N_1 \geq 0 \) such that
\[
\|C_k\| \leq \frac{\varepsilon}{3} \quad \text{for all } k \geq N_1.
\]

Since the algorithm does not terminate,
\[
\|W_k^T \nabla \ell_k\| + \|C_k\| > \varepsilon.
\]
Therefore
\[
\|W_k^T \nabla \ell_k\| \geq \frac{2}{3} \varepsilon \quad \text{for all } k \geq N_1.
\]

There also exists \( N_2 > 0 \) such that for all \( k \geq N_2 \),
\[
\|C_k\| \leq \frac{\varepsilon}{3\omega K_3 K_4}.
\] (7.12)

Now for all \( k \geq \max\{N_1, N_2\} \),
\[
\|W_k^T \nabla q_k(s_k^p)\| = \|W_k^T (\nabla \ell_k + H_k s_k^p)\|
\geq \|W_k^T \nabla \ell_k\| - \|W_k^T \|H_k\| \|s_k^p\|
\geq \frac{2}{3} \varepsilon - K_3 K_4 \|C_k\|
\geq \frac{2}{3} \varepsilon - K_3 K_4 \frac{\varepsilon}{3\omega K_3 K_4}
\geq \frac{1}{3} \varepsilon.
\] (7.13)

The inequality (7.14) is obtained by substituting (7.12) in (7.13). Let
\[
G_k = \{ s : \|s\| \leq \delta_k, \quad q_k(s_k^p + s) - q_k(s_k^p) \leq \sigma[ q_k(s_k^p + s_k^c) - q_k(s_k^p) ] \},
\]
for some \( 0 < \sigma \leq 1 \).
Since $s_k \in \mathcal{G}_k$,

$$q_k(s_k^w) - q_k(s_k^w + s_k) \geq \frac{1}{2} \| W_k^T \nabla q_k(s_k^w) \| \min \{ K_7 \| W_k^T \nabla q_k(s_k^w) \|, \delta_k \}$$

$$\geq \frac{\varepsilon}{6} \min \left\{ K_7 \frac{\varepsilon}{3}, \sqrt{1 - r^2} \delta_k \right\}$$

$$\geq \frac{\varepsilon}{6} \min \left\{ K_7 \frac{\varepsilon}{3 \delta_{\max}}, \sqrt{1 - r^2} \right\} \delta_k,$$  \hspace{1cm} (7.15)

for all $k \geq N_1$.

Now, in the left-hand side we have

$$q_k(s_k^w) - q_k(s_k^w + s_k) = -\frac{1}{2} s_k^T H_k s_k - \nabla q_k(s_k^w)^T s_k$$

$$\leq \frac{1}{2} \| H_k \| \| s_k \|^2 + \| \nabla q_k(s_k^w) \| \| s_k \|$$

$$\leq \left[ \frac{1}{2} \| H_k \| \| s_k \| + \| H_k \| \| s_k^w \| + \| \nabla \ell_k \| \right] \| s_k \|$$

$$\leq \left[ \frac{1}{2} K_3 \delta_{\max} + K_3 K_4 \| C_k \| + K_1 \right] \| s_k \|$$

$$\leq \left[ \frac{1}{2} K_3 \delta_{\max} + K_3 K_4 \frac{\varepsilon}{3} + K_1 \right] \| s_k \|.$$  \hspace{1cm} (7.16)

Then using (7.15), we have for all $k \geq \max \{ N_1, N_2 \}$,

$$\left[ \frac{1}{2} K_3 \delta_{\max} + K_3 K_4 \frac{\varepsilon}{3} + K_1 \right] \| s_k \| \geq \frac{\varepsilon}{6} \min \left\{ K_7 \frac{\varepsilon}{3 \delta_{\max}}, \sqrt{1 - r^2} \right\} \delta_k.$$

That is,

$$\| s_k \| \geq K_{16} \delta_k,$$  \hspace{1cm} (7.17)

where

$$K_{16} = \frac{\frac{\varepsilon}{6} \min \left\{ K_7 \frac{\varepsilon}{3 \delta_{\max}}, \sqrt{1 - r^2} \right\}}{\frac{1}{2} K_3 \delta_{\max} + K_3 K_4 \frac{\varepsilon}{3} + K_1}.$$

Now, from Lemma 7.3, we know that $\delta_k \geq \delta_*$. Then in (7.17) we have that for all $s_k \in \mathcal{G}, k \geq \max \{ N_1, N_2 \}$,

$$\| s_k \| \geq K_{16} \delta_*.$$

Therefore $\| s_k \|$ cannot converge to zero. \qed
7.3 The global convergence theorem

The objective of this section is to prove that under the standard assumptions, the general nonlinear programming algorithm generates a sequence of iterates \( \{x_k\} \), that has at least a subsequence which converges to a stationary point of the problem.

**Theorem 7.2**

Under the standard assumptions,
\[
\liminf_{k \to \infty} \left[ \| \nabla f_k \|^2 + \| C_k \| \right] = 0.
\]

**Proof**

Suppose that the algorithm does not terminate. The proof follows by contradiction. Let us begin by assuming that \( \{ \| C_k \| \} \) does not converge to 0. Then there exists \( \tau > 0 \) such that for all \( k, \| C_k \| \geq \tau \).

First, we are going to find an iterate, say \( x_k \), and a neighborhood \( U \) of \( x_k \) such that \( L(x; \rho) \) is decreased by a positive quantity for infinitely many iterates in \( U \). This will contradict the hypothesis that \( L(x; \rho) \) is bounded below on the domain ID.

Observe that for \( x \in \text{ID} \),
\[
\| C(x) \| = \| C(x) - C(x_k) + C(x_k) \| \\
\geq \| C(x_k) \| - \| C(x) - C(x_k) \|.
\]

Now,
\[
\| C(x) - C(x_k) \| = \left\| \int_0^1 \nabla C(x + t(x - x_k))(x - x_k) \, dt \right\| \\
\leq \int_0^1 \| \nabla C(x + t(x - x_k))(x - x_k) \| \, dt \\
\leq \nu_3 \| x - x_k \|.
\]

Then
\[
\| C(x) \| \geq \| C(x_k) \| - \nu_3 \| x - x_k \|.
\]

Let us define
\[
\sigma = \frac{\| C(x_k) \|}{2\nu_3},
\]
and let \( U(x_k, \sigma) \) be a neighborhood of \( x_k \) of radius \( \sigma \). Then for all \( x \in U(x_k, \sigma) \), we have
\[
\| C(x) \| \geq \frac{1}{2} \| C(x_k) \|. \tag{7.18}
\]
Now let us consider \( x_j \in \mathcal{U}(x_{\hat{k}}, \sigma), \ j \geq \hat{k} \).

Lemma 6.1, Corollary 4.1, the inequality (7.18) and the fact that \( \|C_k\| \geq \tau \) and \( \delta_j \geq \delta_* \) allow us to write the following inequalities:

\[
\begin{align*}
\text{Pred}_j & \geq \frac{\rho_j}{2} \left[ \|C_j\|^2 - \|\Theta_j\|^2 \right] \\
& \geq \frac{K_8}{2} \|C_j\| \min \{ K_6 \|C_j\|, r \delta_j \} \\
& \geq \frac{K_8}{4} \|C_k\| \min \{ K_6 \|C_k\|, r \delta_j \} \\
& \geq \frac{K_8 \varepsilon}{12} \min \{ K_6 \tau, r \delta_j \} \\
& \geq \frac{K_8 \varepsilon}{12} \min \{ K_6 \tau, r \delta_* \}. 
\end{align*}
\]

(7.19)

Then we have, for all \( j \geq \hat{k} \),

\[
\text{Pred}_j \geq \min \{ \kappa_1, \kappa_2 \delta_j \}
\]

(7.20)

where \( \kappa_1 \) and \( \kappa_2 \) do not depend on the iterates.

Since an acceptable step can be found in a finite number of iterations and

\[
\text{Ared}_j \geq \eta_1 \text{Pred}_j \\
\geq \eta_1 \min \{ \kappa_1, \kappa_2 \delta_j \} \\
\geq \eta_1 \min \{ \kappa_1, \kappa_2 \delta_* \} = K_{17},
\]

That is,

\[
\mathcal{L}_j - \mathcal{L}_{j+1} \geq K_{17} \quad \text{for all } x_j \in \mathcal{U}(x_{\hat{k}}, \sigma), \ j \geq \hat{k},
\]

so there are infinitely many iterates for which \( \mathcal{L} \) is decreased by a positive quantity.

Hence we have a contradiction of the assumption that \( \mathcal{L}(x; \rho) \) is bounded below on \( \text{ID} \) and it means that \( x_{\hat{k}} \) must leave the neighborhood.

Hence, let us consider the iterates outside the neighborhood \( \mathcal{U}(x_{\hat{k}}, \sigma) \). Let \( i \) be

\[
i = \min \{ j : j \in \{ k_i \}, \ j > \hat{k}, \ x_{j+1} \notin \mathcal{U}(x_{\hat{k}}, \sigma) \}.
\]

Since \( x_{i+1} \neq x_{\hat{k}} \), there exists at least one acceptable step from iteration \( \hat{k} \) through iteration \( i \), that is the set

\[
\mathcal{A}_i = \{ p : s_p \text{ is acceptable, } p = \hat{k}, ..., i \} \neq \emptyset.
\]
For all $p \in A_i$,

$$
\mathcal{L}_k - \mathcal{L}_{i+1} = \sum_{p=k}^{i} (\mathcal{L}_p - \mathcal{L}_{p+1})
= \sum_{p=k}^{i} Ared_p \\
\geq \sum_{p=k}^{i} \eta_1 Pred_p
$$

Now, by using (7.20) and the fact that $\delta_k \geq \delta_*$ we have

$$
\mathcal{L}_k - \mathcal{L}_{i+1} \geq \eta_1 \sum_{p=k}^{i} \min \{ \kappa_1, \kappa_2 \delta_p \} \\
\geq \eta_1 \sum_{p=k}^{i} \min \{ \kappa_1, \kappa_2 \delta_* \} \\
= K_{18}.
$$

Then again we have that there are infinitely many iterations in which $\mathcal{L}(x; \rho)$ is
decreased by a positive quantity, which contradicts the assumption that $\mathcal{L}(x; \rho)$ is
bounded below on ID. Thus the assumption $\|C_k\| \geq \tau$ can not hold. Therefore there
exists a subsequence $\{k_i\}$ such that

$$
\lim_{k_i \to \infty} \|C_{k_i}\| = 0.
$$

Now let us suppose that $\|W_k^T \nabla \ell_k\| \geq \tau$.

Now, since $\|C_{k_i}\|$ goes to zero and the sequence of trust region radii is bounded away
from zero, there exists an index $N_1 > 0$ such that for all $k_i \geq N_1$, $\|C_{k_i}\| \leq \alpha \delta_{k_i}$, with $\alpha$ as in Lemma 5.10. Then from (5.24) we have that the predicted decrease

$$
Pred_{k_i} \geq \frac{1}{12} \frac{\varepsilon}{\delta_{k_i}} \min \left\{ \frac{1}{2} \delta_{k_i}, \frac{K_7 \varepsilon}{3} \right\}
$$

$$
\geq \frac{\varepsilon}{12} \min \left\{ \delta_{k_i}, K_7 \varepsilon \right\}.
$$

On the other hand, since the general algorithm is well defined at each iteration
we can find an acceptable step, then for all $k_i \geq N_1$,

$$
Ared_{k_i} \geq \eta_1 Pred_{k_i}.
$$
\[ \geq \eta_1 \frac{\varepsilon}{12} \min \left\{ \delta_s, K_s \frac{\varepsilon}{3} \right\} \]
\[ = K_{19}. \]

Then for all \( k_i \geq N_3 \),
\[ \mathcal{L}_{k_i} - \mathcal{L}_{k_{i+1}} \geq K_{19}. \]

This again contradicts the fact that \( \mathcal{L}(x; \rho) \) is bounded below on \( \mathbb{ID} \), since there are infinitely many \( x_{k_i} \) for which \( \mathcal{L}(x; \rho) \) is decreased by a positive quantity. Thus \( \| \nabla_{k_i} \mathcal{L}_{k_i} \| \geq \tau \) cannot hold, so
\[ \lim_{k_i \to \infty} \| \nabla_{k_i} \mathcal{L}_{k_i} \| = 0. \quad (7.26) \]

Then from (7.23) and (7.26) we have
\[ \liminf_{k \to \infty} \left( \| \nabla_{k} \mathcal{L}_{k} \| + \| C_{k} \| \right) = 0. \]

This completes the proof. \( \square \)
Chapter 8

Concluding remarks

We have established a global convergence theory for a class of nonlinear programming algorithms for the smooth problem with equality constraints. The class includes algorithms based on the full space approach like the Celis-Dennis-Tapia subproblem and on tangent space approach like algorithms based on the Vardi subproblem. The family is characterized by generating steps that satisfy very mild conditions on the normal and tangential components. The normal component satisfies a fraction of Cauchy decrease condition on the quadratic model of the linearized constraints and the tangential component satisfies a fraction of Cauchy decrease condition on the quadratic model of the Lagrangian function associated with the problem, reduced to the tangent space of the constraints. The Fletcher exact penalty function was chosen as a merit function. The scheme for updating the penalty parameter is the one proposed by El-Alem in 1991 [9].

The algorithm was proved to be well-defined, in the sense that at each iterate an acceptable step can be found after solving a finite number of trust region subproblems. Because of the properties of the step, we were able to prove that the sequence of trust region radii is bounded away from zero. This result together with the way that the penalty parameter is chosen allowed us to prove that the sequence of penalty parameter is nondecreasing and is increased only finitely many times. The global convergence analysis was constructed on these results.

We have also suggested an algorithm of the class that should work quite well for large problems. The algorithm is a generalization of the Steihaug-Toint dogleg method for the unconstrained case, via a tangent space subproblem.

8.1 Current and future work

For future work, there are some questions that we would like to answer:

- Currently, the local analysis of this class of algorithms is being studied and a preliminary implementation of the algorithm based on the CRG/Steihaug-Toint
via a Vardi subproblem has to be completed. An efficient implementation should be based on the right selection of the submatrix $B$ in the CRG-algorithm.

- A related question that has to be looked at is the search of preconditioners. We believe that the reducer matrix $W$ should play a role in that search.

- Another research topic is the choice of a Lagrange multiplier. For the CRG/Steinhaug-Toint via a Vardi subproblem, rather than a projected Lagrange multiplier we are considering in a reduced one, $\lambda = -B^{-1}\nabla f$. This could match better with the reducer matrix $W$.

- This theory is developed for the equality constrained case, but it can be applied to the general case, by one of the strategies known as EQP and IQP. Let us remember that in the EQP strategy the choice of the active set is made outside the algorithm that determines the step while in the IQP strategy, that choice is made inside the procedure that determines the step. Since the active set may change at each iteration, the choice of the submatrix $B$, will be strongly affected. Certainly, this is an important topic that deserves to be investigated.
Appendix A

An algorithm for computing a trial step

Here we write in more detail the algorithm for computing a trial step, described in page 45.

Algorithm 4.4

Given $x_c \in \mathbb{R}^n$ and $\delta_c > 0$,

I. FEASIBILITY:

1) If $x_c$ is feasible go to II.

2) (* Determine $s_c^u$ by using Algorithm 4.1, the CRG-algorithm on the problem

\[
\begin{aligned}
\text{minimize} & \quad \frac{1}{2} s^T s \\
\text{subject to} & \quad \nabla C_c^T (s - s_{c}^{\text{ff}}) = 0.
\end{aligned}
\]

starting from $s_{c}^{\text{ff}}$, where $s_{c}^{\text{ff}} = (-B_c^{-1}C_c, 0)^T$ or $s_{c}^{\text{ff}} = s_{cra}^{\text{ff}}$ with \[\|s_{cra}^{\text{ff}}\| \leq r\delta_c. \]

step 0: (Initialization)

Set $s_0 = s_{c}^{\text{ff}}$.
Set $\bar{r}_0 = -W^T(H\hat{s}_0 + h)$.
Set $\bar{d}_0 = \bar{r}_0$.
Set $i = 0$.

step 1: (Determine the next iterate)

Compute $\gamma_i = d_i^T \bar{H}d_i$.
Compute $\alpha_i = \frac{\|\bar{r}_i\|^2}{\gamma_i}$.
Set $\hat{s}_{i+1} = \hat{s}_i + \alpha_i d_i$.

step 2: (Determine the next direction)

Compute $\hat{r}_{i+1} = \bar{r}_i - \alpha_i \bar{H}d_i$.
If $\|\hat{r}_{i+1}\| \neq 0$ then

\[
\begin{aligned}
\text{II.} & \quad \text{SOLUTION} \quad \text{IF} \quad \hat{r}_{i+1} \\
\text{III.} & \quad \text{CORE} \quad \text{ELSE}
\end{aligned}
\]
Compute $\beta_i = \frac{||r_i||^2}{||r_i||^2}$.
Set $d_{i+1} = r_{i+1} + \beta_i d_i$.
Set $i = i + 1$ and go to step 1.
otherwise set $s_e = s_{i+1}$ and terminate.

II. MINIMIZATION:

(* Find $s_e$ by applying Algorithm 4.3, the CRG/Steinhaug-Toint algorithm, to

\[
\begin{aligned}
\text{minimize } & q_c(s) \\
\text{subject to } & \nabla C^T(s - s^e) = 0 \\
& \|s\| \leq \delta_e. 
\end{aligned}
\]

* )

step 0: (Initialization)
Set $\hat{s}_0 = s^e$.
Set $r_0 = -W_c^T(H_c s^e + \nabla f_c)$.
If $r_0 = 0$ then terminate.
Set $d_0 = r_0$.
Set $i = 0$.

step 1: Compute $\gamma_i = d_i^T H_c d_i$.
If $\gamma_i > 0$ then go to step 2:
otherwise (* $d_i$ is a direction of negative or zero curvature *)
compute $\tau > 0$ such that $\|\hat{s}_i + \tau d_i\| = \delta_e$.
Set $s_e = \hat{s}_i + \tau d_i$ and terminate.

step 2: Compute $\alpha_i = \frac{||p_i||^2}{\gamma_i}$.
Set $p_{i+1} = p_i + \alpha_i d_i$.
If $||p_i|| < \delta_e$ go to step 3:
otherwise (* the step is too long, take the dogleg step *)
compute $\tau > 0$ such that $\|\hat{s}_i + \tau d_i\| = \delta_e$.
Set $s_e = \hat{s}_i + \tau d_i$ and terminate.

step 3: Compute $r_{i+1} = r_i - \alpha_i H_c d_i$.
If $\frac{||r_{i+1}||}{||\nabla f_c||} \leq \xi$, ($0 < \xi_e \leq \xi < 1$) then
set $s_e = p_{i+1}$ and terminate.
**step 4:** Compute $\beta_i = \frac{\|r_{i+1}\|^2}{\|r_i\|^2}$.
Set $d_{i+1} = r_{i+1} + \beta_i d_i$.
Set $i = i + 1$ and go to **step 1**:
Bibliography


