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 robust choice of the Lagrange multipliers in the successive quadratic programming method

Cores-Carrera, Debora, Ph.D.

Rice University, 1994
A Robust Choice of the Lagrange Multipliers in the Successive Quadratic Programming Method

by

Debora Cores

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree

Doctor of Philosophy

Approved, Thesis Committee:

Richard A. Tapia, Chairman
Noah Harding Professor of Computational and Applied Mathematics

John E. Dennis, Jr.,
Noah Harding Professor of Computational and Applied Mathematics

Keith Cooper,
Associate Professor of Computer Science

Houston, Texas

September, 1993
Abstract

A Robust Choice of the Lagrange Multipliers in the Successive Quadratic Programming Method

by

Debora Cores

We study the choice of the Lagrange multipliers in the successive quadratic programming method (SQP) applied to the equality constrained optimization problem.

It is known that the augmented Lagrangian SQP-Newton method depends on the penalty parameter only through the multiplier in the Hessian matrix of the Lagrangian function. This effectively reduces the augmented Lagrangian SQP-Newton method to the Lagrangian SQP Newton method where only the multiplier estimate depends on the penalty parameter. In this work, we construct a multiplier estimate that depends strongly on the penalty parameter and we derive a choice for the penalty parameter so that the Hessian matrix, restricted to the null space of the constraints, is positive definite and well conditioned. We demonstrate that the SQP-Newton method with this choice of Lagrange multipliers is locally and q-quadratically convergent.
Acknowledgments

First, I would like to thank my husband Marcos. Without his support, encourage and love I would never have made it this far.

I would like to thank the members of my committee: Professors Richard Tapia, John Dennis and Keith Cooper, for their help and interest during the course of this work. I am specially grateful to Professor Richard Tapia for his valuable comments and suggestions and also for extending his warmth and friendship.

I would like to thank all my friends in the department for their help and support during my stay at Rice University. I am also grateful to INTEVEP (Research Center for the Venezuelan Oil Industry) for financial support during part of my graduate career at Rice University.

Finally, I wish to dedicate this work to my mother Isabel and my family for their unfailing love and affection.
Contents

Abstract ii
Acknowledgments iii
List of Tables vi

1 Introduction 1

2 Preliminaries 4

3 Historical Overview 8
  3.1 Sequential Unconstrained Minimization Techniques 8
     3.1.1 The Penalty Function Method 8
     3.1.2 The Augmented Lagrangian Multiplier Method 10
  3.2 The Successive Quadratic Programming Method 13

4 A Robust Choice of The Lagrange Multipliers 15

5 The One Constraint Case ($m = 1$) 23
  5.1 Algorithm for the Penalty Parameter 28
  5.2 An Explicit Expression for the Penalty Parameter 29
  5.3 Numerical Results for $m=1$ 31

6 The $m > 1$ Constraint Case 36

7 Local and q-Quadratic convergence 41

8 Numerical Results 48
  8.1 Implementational Issues 48
  8.2 Numerical Results 49

9 Concluding Remarks 60
Bibliography
# Tables

5.1 Number of starting points for which the SQP-Newton method converges, (indefinite reduced hessian allowed). 34

5.2 Number of starting points for which the SQP-Newton method converges, (indefinite reduced Hessian not allowed). 34

5.3 SQP-Newton method for problem #60. 35

5.4 SQP-Newton method for problem #60. 35

8.1 Number of starting points for which the SQP-Newton method converges, (indefinite reduced Hessian allowed). 51

8.2 Number of starting point for which the SQP-Newton method converges, (indefinite reduced Hessian allowed). 52

8.3 Number of starting points for which the SQP-Newton method converges, (indefinite reduced Hessian allowed). 53

8.4 Number of starting points for which the SQP-Newton method converges, (indefinite reduced Hessian not allowed). 54

8.5 Number of starting points for which the SQP-Newton method converges, (indefinite reduced Hessian not allowed). 55

8.6 Number of starting points for which the SQP-Newton method converges, (indefinite reduced Hessian not allowed). 56

8.7 Number of starting points for which the SQP-Newton method converges. 57

8.8 Number of starting points for which the SQP-Newton method converges. 58

8.9 Number of starting points for which the SQP-Newton method converges. 59
Chapter 1

Introduction

In this work we study the equality constrained nonlinear optimization problem in the successive quadratic programming framework (SQP-Newton method). The SQP-Newton method requires an approximation of the Lagrange multipliers and the solution to an equality constrained quadratic programming problem at each iteration. The objective function of this quadratic problem is a quadratic approximation of the Lagrangian function (Lagrangian SQP-Newton method). The solution of this quadratic problem may not be unique or may not exist if the Hessian matrix of the Lagrangian function is not positive definite on the null space of the constraints. Moreover, even when the solution of this quadratic problem exists, it may not be reliable if the Hessian matrix is not well conditioned. Several decades ago it became fashionable, promoted by the work of Hestenes in 1969, to work with the augmented Lagrangian instead of the Lagrangian. Many researches considered the augmented Lagrangian SQP-Newton method instead of the Lagrangian SQP Newton method. The objective function of the quadratic problem in the augmented Lagrangian SQP-Newton method is a quadratic approximation of the augmented Lagrangian function. Clearly, in this case, we not only require an approximation to the Lagrange multipliers but a value of the penalty parameter. However, the choice of the penalty parameter turned out to be an extremely delicate issue since the augmented Lagrangian SQP-Newton method is quite sensitive to this choice. Little success, if any, was obtained in this direction and researches eventually abandoned the use of the augmented Lagrangian and returned to the standard Lagrangian (choice of zero for the penalty parameter).

It is known that the augmented Lagrangian SQP-Newton method depends on the penalty parameter only through the multiplier in the Hessian matrix of the Lagrangian function. This effectively reduces the augmented Lagrangian SQP-Newton method to the Lagrangian SQP-Newton method where only the multiplier estimate depends on the penalty parameter. The objective of the current work is to derive a choice for the penalty parameter so that the Hessian matrix, restricted to the null space of the
constraints, is positive definite and well conditioned. Moreover, we desire that this choice of the penalty parameter retains the local and q-quadratic convergence of the Lagrangian SQP-Newton method. This work has similarities to Tapia [21] where he used the penalty parameter to obtain effective BFGS and DFP updates for equality constrained optimization in the SQP framework. In that application it turned out that by using structure the penalty parameter cancelled out everywhere in the SQP formulation, except in the scale of the secant method in question. This feature allowed for a successful BFGS and DFP theory eventhough the matrices being approximated were only positive definite on an appropriate subspace.

In this work, we suggest computing the penalty parameter as the solution of an optimization problem which guarantees that the second-order sufficiency conditions for this quadratic programming subproblem hold and the reduced Hessian matrix of the quadratic approximation to the Lagrangian is well conditioned. In this way, the solution of the quadratic problem will be unique and reliable. Our optimization problem for the penalty parameter is a constrained optimization problem whose objective function involves an approximation to the Byrd and Nocedal measure function [4]. The Byrd and Nocedal function is a strictly convex function over the set of symmetric and positive definite matrices and it is uniquely minimizer over this set by the identity matrix. Using the sufficiency condition, given by Tarazaga in [27] and [28], for a symmetric matrix to be positive definite, we restrict the minimization of this objective function to a subset of the set of positive definite matrices.

The numerical results obtained by using this new approach for computing the Lagrange multipliers are encouraging. It was possible to achieve convergence in many examples, where the SQP-Newton method with well-known choices for the Lagrange multipliers (the least-squares multiplier estimate [8], the Miele-Cragg-Levy multiplier [13] and the multiplier associated with the solution of the quadratic program (see Tapia [26] ) did not produce iteration sequences which converged. A significant part of this work is the study of the effectiveness and the robustness of this new choice of Lagrange multiplier estimate in the SQP-Newton framework. Hence, in our numerical comparison we do not embed the SQP-Newton method in a globalization strategy. Our reason is that we feel that good global behavior of the local method speaks strongly to the effectiveness of our multiplier choice.

This work is presented in the following manner. In Chapter 2 we present preliminaries. In Chapter 3 we present a historical overview of the methods used for solving the equality constrained optimization problem, including their advantages
and disadvantages. In Chapter 4 we present our method for computing the penalty parameter in our formula for the Lagrange multiplier. In Chapter 5 we present the optimization problem that we solve at each iteration of the SQP-Newton method in order to obtain the penalty parameter for the case of one constraint. The penalty parameter obtained as the solution of this optimization problem guarantees that the reduced Hessian of the Lagrangian function satisfies the sufficiency conditions for the quadratic subproblem. In Chapter 6 we present an extension of the optimization problem given in Chapter 5 to the case of more than one constraint. In Chapter 7 we establish the local and q-quadratic convergence of the SQP-Newton method with this new choice of the Lagrange multipliers. In Chapter 8 we present some numerical results using our new choice of Lagrange multipliers in the SQP-Newton framework. Finally, in Chapter 9, we present some concluding remarks and discuss future work.
Chapter 2

Preliminaries

In this work, we consider the nonlinear equality constrained optimization problem:

\[
\text{(EQ)} \equiv \begin{cases} 
\text{minimize} & f(x) \\
\text{subject to} & h(x) = 0
\end{cases} \tag{2.1}
\]

where \( f : \mathbb{R}^n \to \mathbb{R} \) and \( h : \mathbb{R}^n \to \mathbb{R}^m \) are assumed to be smooth nonlinear functions and \( m \leq n \). The Lagrangian functional associated with problem (EQ) is

\[
\ell(x, \lambda) = f(x) + h(x)^T \lambda; \tag{2.2}
\]

the augmented Lagrangian functional associated with problem (EQ) is

\[
L(x, \lambda, c) = f(x) + h(x)^T \lambda + \frac{c}{2} h(x)^T h(x); \quad \text{and} \tag{2.3}
\]

the \( l_2 \)-penalty function associated with problem (EQ) is

\[
P(x, c) = f(x) + \frac{c}{2} h(x)^T h(x). \tag{2.4}
\]

In the above \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T \) is called the Lagrange multiplier and the parameter \( c \in \mathbb{R}, c \geq 0 \) is called the penalty parameter.

The vector \( x \in \mathbb{R}^n \) is said to be a regular point of problem (EQ) if the set

\[
\{ \nabla h_1(x), \nabla h_2(x), \ldots, \nabla h_m(x) \}
\]

is linearly independent. If \( x_\ast \in \mathbb{R}^n \) is a regular point, \( f \in C^2(\mathbb{R}^n) \), and \( h \in C^2(\mathbb{R}^n) \), then necessary conditions for \( x_\ast \) to be a local solution of problem (EQ) are:

there exists \( \lambda_\ast \in \mathbb{R}^m \) such that

\[
\nabla_x \ell(x_\ast, \lambda_\ast) = 0 \tag{2.5}
\]

\[
h(x_\ast) = 0, \tag{2.6}
\]
and

$$z^T \nabla^2_x \ell(x_*, \lambda_*) z \geq 0$$

(2.7)

for all $z \in \mathbb{R}^n$, satisfying

$$z^T \nabla h(x_*) = 0.$$  

(2.8)

The conditions (2.5) and (2.6) are known as the first-order necessary conditions and conditions (2.7) and (2.8) are known as the second-order necessary conditions. Sufficient conditions for $x_*$ to be an isolated local solution of problem (EQ) are:

there exists $\lambda_* \in \mathbb{R}^m$ satisfying (2.5) and (2.6) and for every nonzero $z \in \mathbb{R}^n$ satisfying

$$z^T \nabla h(x_*) = 0,$$

(2.9)

we have that

$$z^T \nabla^2_x \ell(x_*, \lambda_*) z > 0.$$  

(2.10)

Conditions (2.9) and (2.10) are equivalent to

$$\eta^T B(x_*)^T \nabla^2_x \ell(x_*, \lambda_*) B(x_*) \eta > 0$$

for all $\eta \neq 0, \eta \in \mathbb{R}^{n-m}$, where $B(x_*)$ is any matrix whose columns form a basis for $\mathcal{N}(\nabla h(x_*)^T)$ (the null space of $\nabla h(x_*)^T$). The vector $x_* \in \mathbb{R}^n$ is said to be a critical point or stationary point of problem (2.1) if there exists a Lagrange multiplier $\lambda_* \in \mathbb{R}^m$ such that $(x_*, \lambda_*) \in \mathbb{R}^{n+m}$ is a solution of the nonlinear system of equations

$$F(x, \lambda) = \nabla \ell(x, \lambda) = \begin{pmatrix} \nabla_x \ell(x, \lambda) \\ h(x) \end{pmatrix} = 0.$$  

(2.11)

The Jacobian matrix of the operator given by (2.11) is

$$\nabla F(x, \lambda) = \begin{pmatrix} \nabla^2_x \ell(x, \lambda) & \nabla h(x) \\ \nabla h(x)^T & 0 \end{pmatrix}.$$  

(2.12)

Throughout this work we will assume that problem (2.1) has a solution $x_*$ with associated Lagrange multiplier $\lambda_*$ and we also assume the standard assumptions for the analysis of Newton’s method:
(A1) \( f, h_i \in C^2(D) \), where \( D \) is some open convex neighborhood of the local solution \( x_* \) of problem (2.1), and \( \nabla^2 f \) and \( \nabla^2 h_i \) are Lipschitz continuous at \( x_* \).

(A2) \( \nabla h(x_*) \) has full rank.

(A3) \( z^T \nabla^2 h(x_*, \lambda_*) z > 0 \) for all \( z \neq 0 \) satisfying \( \nabla h(x_*)^T z = 0 \).

It is well known that if \( x_* \) is a local solution with associated multiplier \( \lambda_* \), then (A2) and (A3) are equivalent to the invertibility of the Jacobian matrix \( \nabla F(x_*, \lambda_*) \) given by (2.12).

Now, applying Newton’s method to (2.11) we obtain the iterative procedure:

\[
\begin{align*}
x_+ &= x + \Delta x_N \\
\lambda_+ &= \lambda + \Delta \lambda_N
\end{align*}
\] (2.13)

where \( \Delta x_N \) and \( \Delta \lambda_N \) are the solutions of the linear system of equations

\[
\begin{pmatrix}
\nabla^2_\ell(x, \lambda) & \nabla h(x) \\
\nabla h(x)^T & 0
\end{pmatrix}
\begin{pmatrix}
\Delta x \\
\Delta \lambda
\end{pmatrix}
=
-
\begin{pmatrix}
\nabla_\ell(x, \lambda) \\
\nabla h(x)
\end{pmatrix}.
\] (2.14)

Today, one of the more popular techniques for solving problem (2.1) is the so-called successive quadratic programming (SQP-Newton) method:

\[
\begin{align*}
x_+ &= x + \Delta x_N \\
\lambda_+ &= \lambda + \Delta \lambda_N
\end{align*}
\] (2.15)

where \( \Delta x_N \) is the solution, and \( \Delta \lambda_N \) is the Lagrange multiplier associated with the solution, of the quadratic program

\[
(QPI) \equiv \begin{cases} 
\text{minimize} & q(\Delta x) = \nabla_\ell(x, \lambda)^T \Delta x + \frac{1}{2} (\Delta x)^T \nabla^2_\ell(x, \lambda) \Delta x \\
\text{subject to} & \nabla h(x)^T \Delta x + h(x) = 0.
\end{cases}
\] (2.16)

It is well known, and not difficult to see, that under the assumption that (QPI) has a solution, the SQP-Newton method is equivalent to Newton’s method applied
to (2.11), i.e., to the iterative procedure described by (2.13). Therefore, the SQP-Newton method is locally and q-quadratically convergent in \((x, \lambda)\) under the standard assumptions (A1)-(A3). For further details on these issues see Fletcher [6] and Bertsekas [1].

The second-order sufficiency conditions for problem (QPI) are

\[ s^T \nabla_x^2 \ell(x, \lambda)s > 0 \quad \text{for all } s \neq 0 \text{ such that} \]

\[ \nabla h(x)^T s = 0. \]

(2.17) \hspace{1cm} (2.18)

Notice that if \( \nabla_x^2 \ell(x, \lambda) \) does not satisfy these sufficiency conditions, then problem (QPI) may not have a solution. Moreover, even if \( \nabla_x^2 \ell(x, \lambda) \) satisfies these conditions, the solution may not be reliable.

When working in the SQP-Newton framework, it is rather common to obtain \( \lambda_+ \), the update of the multiplier \( \lambda \), not as Newton’s method dictates, i.e.,

\[ \lambda_+ = \lambda + \Delta \lambda_N, \]

where \( \Delta \lambda_N \) is obtained from the solution of (QPI), but according to some other strategy. Various alternative strategies can be found in the literature. The primary goal of this research is to present a strategy for obtaining an update multiplier \( \lambda_+ \) so that \( \nabla_x^2 \ell(x_+, \lambda_+) \) is well conditioned and satisfies the appropriate second order sufficiency conditions when used in the (QPI) subproblem. We also desire that local and q-quadratic convergence is maintained. Clearly, such an objective, in full generality, is not possible. However, we expect to contribute significantly in this direction.
Chapter 3

Historical Overview

In this chapter we will describe some of the methods used for solving problem (2.1). First, we study several sequential unconstrained minimization techniques for solving problem (2.1). Then we present the SQP-Newton method which can be viewed as a local technique to solve problem (2.1). The presentation of this material follows Tapia’s notes for constrained optimization [20].

3.1 Sequential Unconstrained Minimization Techniques

In the literature we find various sequential unconstrained minimization techniques for solving problem (2.1):

3.1.1 The Penalty Function Method

The basic idea of the penalty function method is to balance the reduction in the objective function with violation of feasibility in order to obtain global convergence (that is, convergence to a local solution from any initial approximation). This leads to the notion of the penalty function as a positive linear combination of $f$ and $h^T h$.

For the equality constrained optimization problem (2.1) the penalty function method using the penalty function (2.4) can be traced back at least as far as Courant, 1943. A further discussion on this topic can be found in Fletcher [6] and Bertsekas [1].

A generic penalty function method can be outlined as follows:

- Choose an initial penalty constant $c_0$
- For $k = 0, 1, 2, \ldots$ until convergence, perform the following steps:
  - Compute
    $$x_k = \arg \min_{x \in \mathbb{R}^n} P(x, c_k)$$
• Update $c_k$.

The function $P(x, c)$ is given by (2.4). We require $\{c_k\}$ to be a sequence of positive constants increasing monotonically to infinity. Observe that the penalty function method is not really an iterative procedure. Specifically, $x_{k+1}$ does not depend on $x_k$ unless the choice of $c_{k+1}$ depends on $x_k$. The following main convergence result for the penalty function method is due to Polyak [15].

**Theorem 3.1** Assume that $x_*$ is a local solution of problem (2.1) and the standard assumptions (A1-A3) hold. Then there exists a constant $\hat{c} > 0$ such that for every $c > \hat{c}$ the penalty function $P(x, c)$ has a locally unique minimizer, say $x(c)$. Furthermore, there exists a constant $M > 0$ such that

$$\|x(c) - x_*\|_2 \leq \frac{M}{c} \quad \text{for all } c > \hat{c}$$

(3.1)

and

$$\|ch(x(c)) - \lambda_*\|_2 \leq \frac{M}{c} \quad \text{for all } c > \hat{c}.$$  

(3.2)

An immediate consequence of Theorem 3.1 is the following result.

**Corollary 3.1** Suppose that the initial penalty parameter in the penalty function method is larger than $\hat{c}$ in Theorem 3.1. Then the penalty function method is convergent if and only if $c$ goes to infinity.

From (3.1) and (3.2) we observe that it is possible to get arbitrarily good accuracy just by choosing the initial penalty parameter sufficiently large. Of course, the numerical conditioning of the problem (condition number of the Hessian matrix of the penalty function) enters in and it is not clear what the optimal value of $c$ should be. This is expressed mathematically by the fact that the Hessian matrix of the penalty function $\nabla^2 cP(x, c) = \nabla^2 \ell(x, ch(x)) + c\nabla h(x)\nabla h(x)^T$ becomes ill-conditioned since the matrix $c\nabla h(x)\nabla h(x)^T$ has rank $m$ and so there are $m$ eigenvalues of $\nabla^2 cP(x, c)$ which approach infinity as $c$ goes to infinity. The remaining eigenvalues are bounded as a consequence of the Courant-Fisher Theorem (see [12]). In general, it is not clear how one should choose the sequence of penalty parameters $c_k$, and this has been a subject of considerable research.
3.1.2 The Augmented Lagrangian Multiplier Method

The multiplier method is based on the augmented Lagrangian and was originally proposed in 1969 by Hestenes [9] and independently in different but equivalent forms by Powell (1969) [16] and Haarhoff and Buys (1970) [8]. The rationale for the multiplier method is to construct a method which is as effective as the penalty function method but does not suffer from the numerical ill-conditioning of the penalty function method, i.e., the penalty parameter need not become infinite.

Definition 3.1 A function $U : \mathbb{R}^{n+m} \to \mathbb{R}^m$ (which may depend on a parameter $c$) with the property that

$$\lambda_* = U(x_*, \lambda_*, c)$$

whenever $(x_*, \lambda_*)$ is a critical point of problem (2.1) is said to be a Lagrange multiplier update formula. Moreover, if $U$ does not depend explicitly on $\lambda$, i.e.

$$\nabla_\lambda U(x, \lambda, c) = 0$$

then $U$ is said to be a Lagrange multiplier approximation formula.

In the literature we find various forms of Lagrange multiplier update formulas (see Tapia [26]), for example:

$$U(x, \lambda, c) = \lambda + ch(x)$$ (3.3)
$$U(x, \lambda, c) = - (\nabla h(x)^T \nabla h(x))^{-1} \nabla h(x)^T \nabla f(x)$$ (3.4)
$$U(x, \lambda, c) = (\nabla h(x)^T \nabla h(x))^{-1} (h(x) - \nabla h(x)^T \nabla f(x))$$ (3.5)
$$U(x, \lambda, c) = \lambda + (\nabla h(x)^T H \nabla h(x))^{-1} h(x)$$ (3.6)
$$U(x, \lambda, c) = (\nabla h(x)^T H \nabla h(x))^{-1} (h(x) - \nabla h(x)^T H \nabla f(x))$$ (3.7)
$$U(x, \lambda, c) = \lambda + (\nabla h(x)^T D \nabla h(x) + A)^{-1} (h(x) - \nabla h(x)^T D \nabla x L(x, \lambda, c))$$ (3.8)

where the matrix $H = \nabla^2_x c(x, \lambda)^{-1}$. and $A, D$ are $m \times m$ and $n \times n$ matrices which may depend on $x, \lambda, c$. The Lagrange multiplier update formula (3.3) was introduced independently by Hestenes [9] and Powell [16]. Formula (3.4) was introduced by Rosen [17] and then, in 1970, independently of the previous three references, Haarhoff and Buys [8] proposed the multiplier method using the Lagrange multiplier update.
formula (3.4). Observe that the update formula (3.4) is the least-squares solution for 
\( \lambda \) of the overdetermined linear system
\[
\nabla_x \ell(x, \lambda) = 0.
\]
The update formula (3.5) is a special case of a class of formulas used by Miele, Cragg 
and Levy [13]. Formula (3.6) was suggested by Buys [3] for use in the multiplier 
method. Formula (3.7) gives the multiplier that would be obtained by applying 
Newton’s method to the nonlinear system (2.11). The Lagrange multiplier formulas 
(3.7) and (3.8) with \( A = 0 \) are special cases of the general theory developed by Tapia 
[22], [25], [23] and [24] for transforming a constrained problem into an unconstrained 
problem of the same dimension.

The multiplier method is basically the iterative procedure:

- Choose an initial multiplier estimate \( \lambda_0 \) and an initial penalty parameter \( c_0 \)
- For \( k = 0, 1, 2, \ldots \) until convergence, perform the following steps:
  - Compute
    \[
    x_k = \arg \min_{x} L(x, \lambda_k, c_k)
    \]
  - Update \( \lambda_k \) according to \( \lambda_{k+1} = U(x_k, \lambda_k, c_k) \)
  - Update \( c_k \)

In the above \( L(x, \lambda, c) \) is the augmented Lagrangian function (2.3).

Bertsekas in [2] generalized Polyak’s Theorem (Theorem 3.1) to include the multi-
plier method in the following manner. As before we are assuming the standard 
conditions (A1-A3) and \( x_* \) is a local solution of the problem (2.1) with associated 
multiplier \( \lambda_* \).

**Theorem 3.2** Let \( S \) be a bounded subset of \( \mathbb{R}^m \) which contains \( \lambda_* \) as 
an interior point. Then there exists a constant \( \hat{c} \) such that for \( c > \hat{c} \) and 
\( \lambda \in S \) the augmented Lagrangian \( L(x, \lambda, c) \) has a locally unique mini-
mizer, say \( x(\lambda, c) \). Furthermore, there exits a constant \( M > 0 \) such that
\[
\|x_* - x(\lambda, c)\|_2 \leq \frac{M}{c} \|\lambda - \lambda_*\|_2 \quad \text{for all } c > \hat{c} \text{ and } \lambda \in S
\]  
(3.9)
\[ \| \lambda - \hat{\lambda}(\lambda, c) \|_2 \leq \frac{M}{c} \| \lambda - \lambda_* \|_2 \quad \text{for all } c > \hat{c} \text{ and } \lambda \in S \]  

(3.10)

where
\[ \hat{\lambda}(\lambda, c) = \lambda + ch(\lambda, c). \]  

(3.11)

As a direct consequence of Theorem 3.2 and Proposition 9.1 of Tapia [26] we obtain the following result.

**Theorem 3.3** For any given initial estimate of the Lagrange multiplier there exist a penalty constant \( \hat{c} > 0 \) such that the multiplier method with fixed penalty constant \( c > \hat{c} \) is locally \( q \)-linearly convergent in \( x \) or in \( \lambda \).

Observe that in the multiplier method the penalty parameter can not be increased arbitrarily fast as it can be in the penalty function method. If it grows too fast, then \( \hat{\lambda}(\lambda, c) \) given by (3.11) will become excessively large (i.e., it will not remain in the set \( S \) in Theorem 3.2) and the convergence may suffer. It is clear that the increase in \( c \) must be balanced with the decrease resulting from a change in \( x \). However, from (3.11) we see that \( q \)-superlinear convergence would result if it were possible to let the sequence of penalty parameters become unbounded. This latter consideration is the subject of the following theorem which can be found in Tapia [26].

**Theorem 3.4** Suppose that the multiplier method with penalty parameter \( \{ c_k \} \) generates a sequence \( \{ x_k \} \) which is convergent. Then the convergence is \( q \)-superlinear in \( \lambda \) if and only if \( c_k \to \infty \).

The analysis of the penalty parameter in the multiplier method is now complete and we can conclude from the previous results that the price one pays for superlinear convergence is a deterioration in numerical conditioning, since again the penalty constant must go to infinity as in the penalty function method.

Initially there was an extensive amount of effort spent on attacking the constrained minimization problem (2.1) via penalty function methods and augmented Lagrangian methods. These two methods suffer from various computational disadvantages and are not entirely satisfactory. Today one of the most popular techniques for solving problem (2.1) is the successive quadratic programming method.
3.2 The Successive Quadratic Programming Method

By the Lagrangian SQP-Newton method we mean the iterative procedure (2.15) given in Chapter 2.

We can find in the literature, e.g., Fletcher [6] the following convergence result.

**Theorem 3.5** If \((x_0, \lambda_0)\) is sufficiently close to \((x_*, \lambda_*)\) and the standard assumptions (A1-A3) hold, then the Newton method on system (2.11) converges q-quadratically in \((x, \lambda)\) to \((x_*, \lambda_*)\). Moreover, the same is true for the SQP-Newton method.

The following classical result is a primary motivation for using the augmented Lagrangian functional in the SQP-Newton method and in other formulations. This result can be found in Tapia [20].

**Theorem 3.6** (Hestenes)

Assume the standard assumptions (A1)-(A3) at \((x_*, \lambda_*)\). Then there exists \(\hat{c} > 0\) such that for all \(c \geq \hat{c}\) the matrix

\[
\nabla^2_x L(x_*, \lambda_*, c) = \nabla^2_x \ell(x_*, \lambda_*) + c \nabla h(x_*) \nabla h(x_*)^T
\]

is positive definite.

This property of the augmented Lagrangian functional and the expectation that the penalty constant could be used to achieve good global behavior motivated many researches in the 1970's to work with the augmented Lagrangian instead of the Lagrangian in SQP-Newton methods, i.e.:

\[
\begin{align*}
x_{k+1} &= x_k + \Delta x_k \\
\lambda_{k+1} &= \lambda_k + \Delta \lambda_k
\end{align*}
\]

(3.12)

where \(\Delta x_k\) is the solution, and \(\Delta \lambda_k\) is the multiplier associated with the solution, of the quadratic program

\[
\text{(QPL)} \equiv \begin{cases} 
\text{minimize} & q(\Delta x) = \nabla_x L(x_k, \lambda_k, c_k)^T \Delta x + \frac{1}{2} (\Delta x^T \nabla^2_x L(x_k, \lambda_k, c_k) \Delta x) \\
\text{subject to} & \nabla h(x_k)^T \Delta x + h(x_k) = 0.
\end{cases}
\]

(3.13)
However, it turned out that the choice of the penalty parameter $c_k$ was an extremely delicate issue since the augmented Lagrangian SQP-Newton method is very sensitive to the choice of the penalty parameter $c_k$. Due to the limited success in obtaining rules for choosing the penalty parameter eventually researches abandoned the use of the augmented Lagrangian function in SQP-Newton methods. We feel that our choice of the multiplier update will add understanding to the issue of choosing the penalty parameter in the augmented Lagrangian SQP-Newton framework.
Chapter 4

A Robust Choice of The Lagrange Multipliers

In this chapter we present a robust choice of the Lagrange multiplier which strongly utilizes the penalty parameter.

Consider the augmented Lagrangian SQP-Newton method. The first-order necessary conditions associated with the quadratic program (3.13) are

\[
\begin{pmatrix}
\nabla_x^2 L(x_k, \lambda_k, c_k) & \nabla h(x_k) \\
\n\nabla h(x_k)^T & 0
\end{pmatrix}
\begin{pmatrix}
\Delta x_k \\
\Delta \lambda_k
\end{pmatrix}
= 
\begin{pmatrix}
-\nabla_x L(x_k, \lambda_k, c_k) \\
-h(x_k)
\end{pmatrix}.
\] (4.1)

It is not difficult to see that (4.1) is equivalent to

\[
\begin{pmatrix}
\nabla_x^2 \ell(x_k, \lambda_k + c_k h(x_k)) & \nabla h(x_k) \\
\n\nabla h(x_k)^T & 0
\end{pmatrix}
\begin{pmatrix}
\Delta x_k \\
\Delta \lambda_k
\end{pmatrix}
= 
\begin{pmatrix}
-\nabla_x \ell(x_k, \lambda_k) \\
-h(x_k)
\end{pmatrix}.
\] (4.2)

From equation (4.1) and the equivalence between (4.1) and (4.2) we observe that:

- The augmented Lagrangian SQP iterate \( x_{k+1} \) depends on the penalty constant \( c_k \) only through the Hessian matrix \( \nabla_x^2 \ell(x_k, \lambda_k + c_k h(x_k)) \).

- The Hessian matrix of the augmented Lagrangian functional is the Hessian matrix of the Lagrangian functional for a particular choice of the Lagrange multiplier when restricted to the subspace \( \{ s \in \mathbb{R}^n : \nabla h(x_k)^T s = 0 \} \), i.e.,

  \[
  B(x_k)^T \nabla_x^2 \ell(x_k, \lambda_k + c_k h(x_k)) B(x_k) = B(x_k)^T \nabla_x^2 L(x_k, \lambda_k, c_k) B(x_k),
  \]

  where \( B(x_k) \) is any matrix whose columns form a basis for \( \mathcal{N}(\nabla h(x_k)^T) \).

- At a solution \( (x_*, \lambda_*) \) we have \( \nabla_x L(x_*, \lambda_*, c) = \nabla_x \ell(x_*, \lambda_*) = 0 \), for any \( c \in \mathbb{R} \).

- The augmented Lagrangian SQP-Newton method with multiplier \( \lambda_k \) is equivalent (i.e., produces the same iterates) to the Lagrangian SQP-Newton method with multiplier \( \lambda_k + c_k h(x_k) \) in the Hessian matrix of the Lagrangian functional.
From these observations, Theorem 3.6, and the previous discussion concerning the various sequential unconstrained minimization techniques we consider using, as a Lagrange multiplier in the Hessian matrix of the Lagrangian function,

$$\lambda_k = U(x_k) + c_k h(x_k)$$ (4.3)

where $U(x_k)$ is a Lagrange multiplier formula which does not depend on $c_k$. Ideally we propose to choose the penalty parameter in (4.3) so that:

(C1) the Hessian matrix of the Lagrangian functional $\nabla_x^2 \ell(x_k, \lambda_k)$ with $\lambda_k$ given by (4.3) is positive definite on the subspace $\{ s \in \mathbb{R}^n : \nabla h(x_k)^T s = 0 \}$.

(C2) the reduced Hessian matrix $B(x_k)^T \nabla_x^2 \ell(x_k, \lambda_k) B(x_k)$ for $\lambda_k$ given by (4.3) is well conditioned.

(C3) the local convergence properties of the Lagrangian SQP-Newton method are maintained.

Therefore, what we want is to determine a choice $c_k$ of penalty parameter that produces a well-posed and well-conditioned subproblem (3.13) and such that the SQP-Newton method with this choice of the Lagrange multiplier (4.3) will also be locally and quadratically convergent under very mild conditions on the penalty parameter sequence $\{ c_k \}$. We hope that as a by-product we will also obtain improved global behavior of the augmented Lagrangian SQP-Newton method.

It is also important to mention that none of the Lagrange multiplier formulas in the literature guarantee that conditions (C1), (C2) and (C3) are satisfied. Our idea is to compute the Lagrange multiplier from (4.3) and obtain $c_k$ as the solution of an optimization problem of the form:

$$\left\{ \begin{array}{l}
\text{minimize} \quad f(c, \rho) = \| (C h(x_k))^T, \rho \|_2^2 \\
\text{subject to} \quad \hat{H}(c, \rho) \in S
\end{array} \right.$$ (4.4)

where:

$c \in \mathbb{R}^m$, $\rho \in \mathbb{R}$ and $C := \text{diag}(c)$,

$\hat{H}(c, \rho) := H(c) + \rho I$,

$H(c) := B(x_k)^T \nabla_x^2 \ell(x_k, U(x_k, \lambda_k) + C h(x_k)) B(x_k)$,
and $S$ is a closed and convex set, contained in the interior of the set of positive definite matrices (SPD). Below we explicitly define the constraint set $S$ considered in our application.

Notice, that at iteration $k$ of the SQP-Newton method, we are associating a distinct penalty parameter $c_k^i \in \mathbb{R}$ to each one of the equality constraints $h_i(x_k)$ instead of associating the same penalty parameter $c_k \in \mathbb{R}$ to each of the $h_i(x_k)$. Therefore, the Lagrange multiplier formula we suggest for use in this work is

$$\lambda_k = U(x_k) + C_k h(x_k).$$  \hspace{1cm} (4.5)

Our next theorem will show that problem (4.4) is well posed in the following sense: for any iterate $x_k \in \mathbb{R}^n$ the eigenvalues of $H(c_k) \in \mathbb{R}^{(n-m)\times(n-m)}$ are invariant with respect to the choice of the matrix $B(x_k)$ that we choose to represent a basis of $\mathcal{N}(\nabla h(x_k)^T)$. In order to prove that problem (4.3) is well posed we need the following lemmas.

**Lemma 4.1** Consider $A, B \in \mathbb{R}^{(n \times n)}$. If $AB = I$, then $BA = I$.

**Proof** The lemma is standard. A proof can be found in Kincaid and Cheney [11].

**Lemma 4.2** Consider $B_1, B_2 \in \mathbb{R}^{(n \times k)}$, $k \leq n$. If each has orthonormal columns which form a basis for the subspace $V \subseteq \mathbb{R}^n$, then there exists an orthogonal matrix $Q \in \mathbb{R}^{(k \times k)}$ such that $B_2 = B_1 Q$.

**Proof** Let $b_i$ be the $i^{th}$ column of the matrix $B_1$ then $b_i \in V$, for $i = 1, 2, \ldots k$. Therefore there exist $q_i \in \mathbb{R}^k$ for each $i = 1, 2, \ldots k$ such that

$$b_i = B_2 q_i \quad \text{for} \quad i = 1, 2, \ldots k.$$  

This implies $B_1 = B_2 Q$, where $Q$ is the matrix whose columns are $q_i$, $i = 1, \ldots, k$. On the other hand, $B_1^T B_1 = Q^T B_2^T B_2 Q$. Since $B_1$ and $B_2$ have orthonormal columns we have $B_2^T B_2 = I$ and $B_1^T B_1 = I$ and thus $Q^T Q = I$. Now, since $Q \in \mathbb{R}^{(k \times k)}$ and $Q^T Q = I$ by Lemma 4.1 we have that $QQ^T = I$ and therefore $Q$ is an orthogonal matrix.
The previous two lemmas allow us to prove the following result.

**Theorem 4.1** Let \( A \in \mathbb{R}^{(n \times n)} \) be a symmetric matrix. Let \( B_1 \) and \( B_2 \in \mathbb{R}^{(n \times k)} \) with \( n \geq k \) be such that each has orthonormal columns and these two sets of columns each form a basis for the subspace \( V \subseteq \mathbb{R}^n \).

Then \( B_1^T A B_1 \) and \( B_2^T A B_2 \) have the same eigenvalues.

**Proof** It is clear that \( B_1^T A B_2 \) and \( B_2^T A B_1 \) are symmetric matrices. Thus, the matrices \( B_1^T A B_2 \) and \( B_2^T A B_1 \) have \( k \) real eigenvalues and the corresponding eigenvectors form an orthonormal set. Suppose \( \lambda_i \in \mathbb{R} \) is any eigenvalue of the matrix \( B_2^T A B_2 \) with corresponding eigenvector \( x_i \in \mathbb{R}^k \). Then,

\[
B_2^T A B_2 x_i = \lambda_i x_i.
\]

By Lemma 4.2, there exists an orthogonal matrix \( Q \in \mathbb{R}^{(k \times k)} \) such that

\[
B_2 = B_1 Q.
\]

Therefore,

\[
(B_1 Q)^T A (B_1 Q) x_i = \lambda_i x_i,
\]

and so,

\[
Q^T B_1^T A B_1 Q x_i = \lambda_i x_i. \tag{4.6}
\]

Multiplying both sides of (4.6) by \( Q \), we obtain

\[
QQ^T B_1^T A B_1 Q x_i = \lambda_i Q x_i.
\]

Since \( QQ^T = I_{k \times k} \) we have that \( B_1^T A B_1 Q x_i = \lambda_i Q x_i \) which implies, \( \lambda_i \) is an eigenvalue of the matrix \( B_1^T A B_1 \) with corresponding eigenvector \( Q x_i \) and establishes the theorem. \( \square \)

The Hessian matrix \( \nabla_2^2 \ell(x_k, U(x_k) + C_k h(x_k)) \) is a symmetric matrix. Thus, from the previous theorem we have that the reduced Hessian matrix

\[
H(c_k) = B(x_k)^T \nabla_2^2 \ell(x_k, U(x_k) + C_k h(x_k)) B(x_k)
\]

will have the same eigenvalues independently of the matrix \( B(x_k) \) we choose to represent the basis of \( \mathcal{N}(\nabla h(x_k)^T) \). Hence, problem (4.4) will be well posed in the sense that the eigenvalues of \( H(c_k) \) are invariant with respect to the choice of the matrix.
$B(x_k)$. We will denote by SPD the collection of $n \times n$ symmetric and positive definite matrices.

Recently, Byrd and Nocedal [4] introduced a measure function which has the flavor of the condition of a matrix $A \in \text{SPD}$. This function was also studied by Fletcher [7] and is defined as:

$$
\Psi_{BN}(A) = \frac{1}{n} (\text{trace}(A) - \ln(\det(A))), \quad \text{for } x \in \text{SPD}.
$$

(4.7)

The Byrd and Nocedal measure function (4.7) satisfies the following properties:

(P1) $\Psi_{BN}(A)$ is a strictly convex function on the set SPD (see [4] and [7]).

(P2) $\Psi_{BN}(A)$ is globally and uniquely minimized by $A = I$ over the set SPD (see [7]).

(P3) $\Psi_{BN}(A) \geq 1$, for any $A \in \text{SPD}$ (see [4]).

(P4) $\Psi_{BN}(A) > \ln(\text{Cond}(A))$, for any $A \in \text{SPD}$

where $\text{Cond}(A)$ denotes the condition number of the matrix $A$. (see [4]).

In some sense, $\Psi_{BN}(A)$ can be considered to be a measure of the closeness between $A$ and the identity matrix.

Now, consider problem (4.4) where the constraint set $S$ is given by

$$
S = \{(c, \rho) \in \mathbb{R}^{m+1} : \Psi_{BN}(\hat{H}(c, \rho)) = \frac{\text{trace}(\hat{H}(c, \rho)) - \ln(\det_+(\hat{H}(c, \rho)))}{n - m} \leq M\},
$$

(4.8)

for some real $M > 1$.

Here, $\det_+$ of a symmetric matrix $A$ is simply defined as

$$
\det_+(A) := \begin{cases} 
\det(A) & \text{if } A \text{ is positive definite} \\
0 & \text{otherwise}
\end{cases}
$$

Our next result will be used to prove that problem (4.4) with $S$ given by (4.8) has a unique solution.

**Lemma 4.3** The constraint set $S$ given by (4.8) is closed and convex.
**Proof** Let \( \hat{c}_1 = (c_1, \rho_1) \) and \( \hat{c}_2 = (c_2, \rho_2) \) be members of \( S \). Then, \( \Psi_{BN}(\hat{H}(\hat{c}_1)) \leq M \) and \( \Psi_{BN}(\hat{H}(\hat{c}_2)) \leq M \).

Now, suppose that \( \hat{H}(\hat{c}_i), \ i = 1, 2 \) is not positive definite then,

\[
\Psi_{BN}(\hat{H}(\hat{c}_i)) > M.
\]

Thus, we have that \( \hat{H}(\hat{c}_i), \ i = 1, 2 \) is positive definite.

Therefore, for any \( \beta \in [0,1] \) we have that

\[
\Psi_{BN}(\hat{H}(\beta \hat{c}_1 + (1-\beta) \hat{c}_2)) = \Psi_{BN}(\beta \hat{H}(c_1) + (1-\beta)\hat{H}(c_2) + \beta \rho_1 I + (1-\beta)\rho_2 I)
\]

\[
= \Psi_{BN}(\beta \hat{H}(c_1) + \rho_1 I) + (1-\beta)(\hat{H}(c_2) + \rho_2 I)
\]

\[
= \Psi_{BN}(\beta \hat{H}(\hat{c}_1) + (1-\beta)\hat{H}(\hat{c}_2)).
\]

Since \( \hat{H}(\hat{c}_1), \hat{H}(\hat{c}_2) \) are positive definite matrices and \( \Psi_{BN} \) is strictly convex function on SPD it follows that

\[
\Psi_{BN}(\beta \hat{H}(\hat{c}_1) + (1-\beta)\hat{H}(\hat{c}_2)) < \beta \Psi_{BN}(\hat{H}(\hat{c}_1)) + (1-\beta)\Psi_{BN}(\hat{H}(\hat{c}_2))
\]

\[
\leq \beta M + (1-\beta)M = M,
\]

which implies \( \beta \hat{c}_1 + (1-\beta)\hat{c}_2 \in S \) and thus, \( S \) is a convex set. By the continuity of the functions \( \Psi_{BN} \) and \( \hat{H} \) it follows that the set \( S \) given by (4.8) is closed. \( \Box \)

**Theorem 4.2** Problem (4.4) where the set \( S \) is given by (4.8) has a unique solution.

**Proof** By Lemma 4.3 \( S \) is a convex and closed set. Moreover, from the definition of \( f \) we have that \( f \) is continuous and uniformly convex on \( S \). Therefore, problem (4.4) has a unique solution. \( \Box \)

The remainder of this work is concerned with finding a numerical approach for solving problem (4.4). Towards this end, we will write problem (4.4) as a barrier penalty function problem

\[
\begin{align*}
\text{minimize} & \quad F(c, \rho) = \|((Ch(x_k))^T, \rho)\|_2^2 + \mu \Psi_{BN}(\hat{H}(c, \rho)) \\
\text{subject to} & \quad \hat{H}(c, \rho) \in \text{SPD}
\end{align*}
\]

for some \( \mu > 0 \) and \( \hat{H} \) is defined immediately following (4.4).
Notice that the barrier function we are considering,

\[ \Psi_{BN}(\hat{H}(c, \rho)) = \frac{1}{n-m} \{ \text{trace}(\hat{H}(c, \rho)) - \ln(\det(\hat{H}(c, \rho))) \} \]

is a smooth convex function, tends to infinity as the matrix \( \hat{H}(c, \rho) \) approaches the boundary of the set of symmetric and semipositive definite matrices (SSPD), and is finite when the matrix \( \hat{H}(c, \rho) \) is in the interior of the set SSPD (i.e., is in SPD).

The following result will be used to show that problem (4.9) has a unique solution.

**Lemma 4.4** The function \( F : \mathbb{R}^{m+1} \to \mathbb{R} \) given in problem (4.9) has the infinity property in SPD, that is, if \( (c, \rho) \) is such that \( \hat{H}(c, \rho) \in \text{SPD} \), then

\[ \lim_{\| (c, \rho) \|_2 \to \infty} F(c, \rho) = \infty, \]

for a fixed vector \( h(x_k) \), with each component \( h_i(x_k) \), \( i = 1, \ldots, m \), different from zero.

**Proof** We know that \( \Psi_{BN}(\hat{H}(c, \rho)) \geq 1 \) if the matrix \( \hat{H}(c, \rho) \) is in SPD. Therefore,

\[
F(c, \rho) = \|((Ch(x_k))^T, \rho)\|_2^2 + \mu \Psi_{BN}(\hat{H}(c, \rho)) \\
\geq \|((Ch(x_k))^T, \rho)\|_2^2 + \mu \geq \|((Ch(x_k))^T, \rho)\|_2^2 \\
\geq \min(h_1(x_k)^2, h_2(x_k)^2, \ldots, h_m(x_k)^2, 1)\| (c, \rho) \|_2^2
\]

Since \( h_i(x_k) \neq 0 \) for \( i = 1, \ldots, m \), we conclude

\[ \lim_{\| (c, \rho) \|_2 \to \infty} F(c, \rho) = \infty. \]

To establish the infinity property for the function \( F \) given in problem (4.9) we required each component of the vector \( h(x_k) \) to be different from zero. This condition will not affect our results because in the case that a component \( h_i(x_k) = 0 \) we do not compute the corresponding penalty vector \( c_k^i h_i(x_k) = 0 \).

In order to state our next result we first point out that we can always find a vector \((c_1, \rho_1) \in \mathbb{R}^{m+1}\) such that the reduced Hessian matrix

\[
\hat{H}(c_1, \rho_1) = H(c_1) + \rho_1 I \in SPD \quad \text{and} \quad \Psi_{BN}(c_1, \rho_1) \leq M, \quad \text{for fixed } M > 1.
\]
Thus, $F(c_1, \rho_1)$ is well defined and the following set is not empty

$$S_1 = \{(c, \rho) \in S : F(c, \rho) \leq F(c_1, \rho_1)\}.$$

**Theorem 4.3** Problem (4.9) has a unique solution.

**Proof** Consider the optimization problem:

$$
\begin{align*}
\begin{cases}
\text{minimize} & F(c, \rho) = \|((Ch(x_k))^T, \rho)\|_2^2 + \mu \Psi_{BN}(\hat{H}(c, \rho)) \\
(c, \rho) & \in S_1
\end{cases}
\end{align*}
$$

(4.10)

The objective function $F$, of problem (4.9), is the sum of two strictly convex functions on SPD. Thus, $F$ is strictly convex on the set $S_1$. Moreover, the set $S_1$ is closed and bounded since $F$ is continuous on $S_1$, the set $S$ is closed (Lemma 4.3) and the function $F$ has the infinity property in SPD (Lemma 4.4). Thus, the optimization problem (4.10) has a unique solution. On the other hand, since $S$ is contained in SPD and $F$ is strictly convex on SPD we have that the optimization problem (4.9) has a unique solution.

**Theorem 4.4** If $(c^*(\mu), \rho^*(\mu))$ solves problem (4.9), then there exists $M(\mu)$ such that $(c^*(\mu), \rho^*(\mu))$ solves problem (4.4) with $M = M(\mu)$.

**Proof** Suppose $(c^*(\mu), \rho^*(\mu))$ solves problem (4.9). Then, $(c^*(\mu), \rho^*(\mu))$ solves problem (4.4) with $M = \Psi_{BN}(\hat{H}((c^*(\mu), \rho^*(\mu)))$.

In the following two chapters we present a constrained optimization problem which is an approximation to the barrier penalty problem (4.9).
Chapter 5

The One Constraint Case ($m = 1$)

In this chapter, we consider problem (2.1) with only one constraint. For this particular case, we construct a constrained optimization problem, that will be solved at each iteration of the SQP-Newton method, to obtain an approximation to the solution of the barrier penalty problem (4.9), with $\rho_k = 0$. Also we present some preliminary numerical results that show that if we use this approximate penalty vector in the Lagrange multiplier (4.3), then we achieve convergence for many cases where the standard SQP-Newton fails.

Suppose we are at the $k^{th}$ iteration of the SQP-Newton method and consider the reduced Hessian matrix $H(c)$ for the one constraint case i.e.,

$$
H(c) = B(x_k)^T \nabla_x \ell(x_k, U(x_k) + ch(x_k))B(x_k)
= B(x_k)^T (\nabla^2 f(x_k) + U(x_k)\nabla^2 h(x_k))B(x_k)
+ ch(x_k)B(x_k)^T \nabla^2 h(x_k)B(x_k).
$$

(5.1)

For any given iterate $x_k \in \mathbb{R}^n$, we have that all the elements in (5.1) are known except the value of the penalty parameter $c$; so for the sake of simplicity, we can write (5.1) as

$$
H(c_k) = A + ch_k D
$$

where,

$$
h_k = h(x_k),
A = B(x_k)^T \nabla^2 (f(x_k) + U(x_k)\nabla^2 h(x_k))B(x_k) \quad \text{and},
D = B(x_k)^T \nabla^2 h(x_k)B(x_k).
$$

For this particular case, we wish to find a constrained optimization problem whose objective function is an approximation to the barrier penalty function given in problem (4.9) when $\rho_k = 0$. First, we characterize the set of minimizers of the Byrd and Nocedal measure function (4.7) over the set SPD. To obtain this characterization we use the following lemma that can be found in Fletcher [7] and Strang [19].
Lemma 5.1  For a nonsingular matrix $A \in \mathbb{R}^{(n \times n)}$, the partial derivative of $\det(A)$ is given by
\[
\frac{d}{da_{ij}} (\det(A)) = (A^{-T})_{ij} \det(A) = (A^{-1})_{ji} \det(A). \tag{5.2}
\]
Moreover,
\[
\frac{d}{da_{ij}} (\ln(\det(A))) = \frac{(A^{-1})_{ji} \det(A)}{\det(A)} = (A^{-1})_{ji}. \tag{5.3}
\]

Theorem 5.1  Consider the optimization problem
\[
\begin{align*}
\begin{cases}
\text{minimize} & \Psi_{BN}(A + c_h D) \\
H(c) & \in \text{SPD}
\end{cases}
\end{align*} \tag{5.4}
\]
If $h_k \neq 0$, then the solution $c_k$ of (5.4) satisfies
\[
\text{trace}(D - H(c_k)^{-1}D) = 0.
\]

Proof  Let us differentiate the function $\Psi_{BN}(H(c_k))$ with respect to the penalty parameter $c_k$
\[
\frac{d}{dc_k} (\Psi_{BN}(H(c_k))) = \frac{d}{dc_k} (\Psi_{BN}(A + c_k h_k D)) \\
= \frac{1}{n - m} \left\{ \frac{d}{dc_k} \left( \text{trace}(A + c_k h_k D) - \ln(\det(A + c_k h_k D)) \right) \right\} \\
= \frac{1}{n - m} \left\{ h_k \text{trace}(D) - \frac{d}{dc_k} (\ln(\det(H(c_k)))) \right\}. \tag{5.5}
\]
On the other hand, by Lemma 5.1 we have that
\[
\frac{d}{dc_k} (\ln(\det(H(c_k)))) = \frac{1}{\det(H(c_k))} \frac{d}{dc_k} (\det(H(c_k))) \\
= \frac{1}{\det(H(c_k))} \left\{ \sum_{i,j=1}^{n} \frac{d}{dc_k} (\det(H(c_k))) \frac{d}{dc_k} (H_{ij}) \right\}. \tag{5.6}
\]
Replacing the derivative of the determinant of the matrix $H(c_k)$ with respect to the element $H_{ij}$ in (5.6) and using Lemma 5.1 we obtain
\[
\frac{d}{dc_k} \left( \ln(\det(H(c_k))) \right) = \frac{1}{\det(H(c_k))} \left\{ \sum_{i,j=1}^{n-m} \det(H(c_k)) H(c_k)_{ji}^{-1} d_{ij} h_k \right\}
\]
\[
= h_k \left\{ \sum_{i,j=1}^{n-m} H(c_k)_{ji}^{-1} d_{ij} \right\}
\]
\[
= h_k \text{ trace}(H(c_k)^{-1} D).
\]

This implies that (5.5) can be written as,
\[
\frac{d}{dc_k} \left( \Psi_{BN}(H(c_k)) \right) = \frac{1}{n - m} \left\{ h_k \text{ trace}(D) - h_k \text{ trace}(H(c_k)^{-1} D) \right\}
\]
\[
= \frac{h_k}{n - m} \left\{ \text{trace}(D - H(c_k)^{-1} D) \right\}.
\]

If $c_k$ is a solution of problem (5.4) and $h_k \neq 0$ we have that $c_k$ satisfies the first-order necessary conditions, that is:
\[
\text{trace}(D - H(c_k)^{-1} D) = 0.
\] (5.7)

From this result we observe that if there exists $c_k$ such that $H(c_k) = I$, then equation (5.7) is satisfied and thus $c_k$ is the unique minimizer of problem (5.4). However, this is not always possible nor desirable and it is not trivial to obtain an expression for $c_k$ from the first-order necessary condition (5.7). Therefore, we consider substituting $H(c_k)^{-1}$ for $H(c_k)$ in (5.7) to obtain
\[
\text{trace}(D - H(c_k) D) = 0.
\] (5.8)

Now, consider the following equation
\[
c_k h_k + \mu \text{trace}(D - H(c_k) D) = 0.
\] (5.9)

The first term on the left-hand side of (5.9) corresponds to the first-order necessary conditions for the minimization of $\|c_k h\|_2^2$ and we are assuming $h_k \neq 0$. Our next result shows that there exits a relation between equation (5.9) and the following optimization problem:
\[
\begin{aligned}
\begin{cases}
\text{minimize} & \phi(c) = \|c h_k\|_2^2 + \mu \|H(c) - I\|_F^2, \\
& c \in \mathbb{R}
\end{cases}
\end{aligned}
\] (5.10)

**Theorem 5.2** A necessary and sufficient condition for \( c_k \) to be a minimizer of problem (5.10) is that \( c_k \) satisfy equation (5.9).

**Proof** Let us consider \( \phi(c) = \|c h_k\|_2^2 + \mu \|H(c) - I\|_F^2 \).

We have that, \( \phi : \mathbb{R} \to \mathbb{R} \) can be written

\[
\phi(c_k) = c_k^2 h_k^2 + \mu \{\text{trace}(A^2) + 2c_k h_k \text{trace}(AD) + c_k^2 h_k^2 \text{trace}(D^2) - 2\text{trace}(A) - 2c_k h_k \text{trace}(D) + (n - m)\}.
\] (5.11)

The necessary conditions for \( c_k \) to be a minimizer of problem (5.10) is,

\[
\phi'(c_k) = 2c_k h_k^2 + \mu \{2h_k \text{trace}(AD) + 2c_k h_k^2 \text{trace}(D^2) - 2h_k \text{trace}(D)\} = 0,
\]

or equivalently,

\[
2c_k h_k^2 + 2\mu h_k \text{trace}((A + c_k h_k D)D - D) = 2c_k h_k^2 + 2\mu h_k \text{trace}(H(c_k) D - D) = 0.
\]

If \( h_k \neq 0 \), and \( \text{trace}(D^2) \neq 0 \) then the necessary condition for \( c_k \) to be a minimizer of problem (5.10) is

\[
c_k h_k + \mu \text{trace}(D - H(c_k) D) = 0.
\] (5.12)

On the other hand, we know that the function \( \phi \) is uniformly convex on \( \mathbb{R} \). Since \( \mathbb{R} \) is a convex and closed set then, equation (5.12) is a necessary and sufficient condition for \( c_k^* \) to be a minimizer of problem (5.10).

Theorem 5.2, Theorem 5.1, and the fact that the minimizer of the Byrd and Nocedal measure function (4.7) over the set SPD is the identity matrix, motivate us to approximate the barrier penalty function \( F \) given in problem (4.9) with the following function:

\[
\phi(c) = \|c h_k\|_2^2 + \mu \|H(c) - I\|_F^2.
\] (5.13)

It is clear, that for the minimizer \( c_k \) of \( \phi \) we can not guarantee that the matrix \( H(c_k) \) is positive definite. For this reason, we need to restrict the minimization of \( \phi \)
to a set contained in SPD. A necessary and sufficient condition for a matrix to be positive definite is that all the eigenvalues of the matrix are positive. To enforce this condition requires considerable numerical computation at every iteration of the SQP-Newton scheme. However, Tarazaga in [27] and [28] presented a sufficiency condition for a matrix $A \in \mathbb{R}^{n \times n}$ to be positive definite. This condition does not require us to know the eigenvalues of the matrix $A$ and it is an easy condition to evaluate. This condition only depends on the elements of the matrix $A$ and is stated as the following theorem (see [28]).

**Theorem 5.3** (Tarazaga)

Let $A \in \mathbb{R}^{n \times n}$. If

$$\text{trace}(A) > (n - 1)^{\frac{1}{2}} \|A\|_F,$$

then $A$ is positive definite.

We propose the use of condition (5.14) as the constraint

$$g(H(c_k)) \equiv \text{trace}(H(c_k)) - (n - m - 1)^{\frac{1}{2}} \|H(c_k)\|_F \geq \alpha,$$

where

$$\alpha = \text{trace}(I) - (n - m - 1)^{\frac{1}{2}} \|I\|_F$$

$$= (n - m) - \sqrt{(n - m)(n - m - 1)} > 0.$$

Observe that inequality (5.15) guarantees that the matrix $H(c_k)$ is positive definite and the equality holds when $H(c_k) = I$, which is, certainly a well conditioned matrix. Thus, the constrained optimization problem we propose to solve at each iteration of the SQP-Newton method to obtain an approximation to the solution $c_k$ of problem (4.9) satisfying objectives (C1) and (C2) is:

$$\begin{cases}
\text{minimize} & \phi(c) = \|ch_k\|^2 + \mu \|H(c) - I\|_F^2 \\
\text{subject to} & \text{trace}(H(c)) - \sqrt{(n - m - 1)} \|H(c)\|_F \geq \alpha.
\end{cases}$$

(5.16)

The following result establishes that, under suitable conditions, problem (5.16) has a unique solution. First, we present some preliminary results. Recall that $H(c) = A + ch_k D$.

**Lemma 5.2** The set $\Omega = \{c \in \mathbb{R} : g(H(c)) \geq \alpha\}$ is closed and convex.
Proof The proof follows from the fact that $H$ and $g$ are continuous and convex.

Lemma 5.3 The function $\phi$ given by (5.16) is a uniformly convex function on $\Omega = \{c \in \mathbb{R} : g(H(c)) \geq \alpha\}$.

Proof The proof of this lemma follows directly from the definition of the function $\phi$.

Theorem 5.4 The optimization problem (5.16) has a unique solution provided the set $\Omega$ is not empty.

Proof By Lemma 5.2 the set $\Omega \subset \mathbb{R}$ is closed and convex. Moreover, by Lemma 5.3 the function $\phi : \Omega \subset \mathbb{R} \to \mathbb{R}$ is continuous and uniformly convex function on $\Omega$. Therefore, problem (5.16) has a unique solution provided $\Omega$ is not empty.

5.1 Algorithm for the Penalty Parameter

The algorithm we propose for obtaining $(c, \rho)$, which is an approximation to the solution $(c_k, \rho_k)$ of problem (4.9), at the $k^{th}$ iteration of the SQP-Newton method, is as follows:

Algorithm 5.5

Given $\mu$ do the following:
If $\hat{H}(0,0) \in \Omega$
   Set $(c_k, \rho_k) = (0,0)$
Else
   Let $c_k^*$ be the solution of the constrained problem (5.16), if it exists. Otherwise, let $c_k^*$ be the solution of the unconstrained problem (5.10).
   If $\hat{H}(c_k^*,0) \in \Omega$
      Take $(c_k, \rho_k) = (c_k^*,0)$
   Else
      Take $(c_k, \rho_k) = (0, \rho_k)$ where $\rho_k$ is computed such that $H(0) + \rho_k I \in \text{SPD}$
End
End
Notice that in Algorithm 5.5 we compute $\rho_k$ only if $H(0) \notin \Omega$ and $H(c_k^*) \notin \Omega$. Moreover, in this case we take $c_k = 0$ and add $\rho_k I$ to the reduced Hessian matrix $H(0)$. This reasoning comes from the fact that $c_k$ obtained in the first steps of the algorithm is not a satisfactory choice for the penalty parameter. Thus, in this case makes better sense to correct the matrix $H(0)$ instead of the matrix $H(c_k)$. In order to compute $\rho_k$ we use the modified Cholesky factorization as presented in Dennis and Schnabel [5]. Also notice that near a solution, $\rho_k = 0$ even if $\dot{H}(0,0) \notin \Omega$ and $\dot{H}(c_k^*,0) \notin \Omega$.

5.2 An Explicit Expression for the Penalty Parameter

We will use Lagrange multiplier theory to find an explicit solution of problem (5.16) for the case $m = 1$.

The Lagrangian function associated with problem (5.16) is

$$\ell(c_k, \gamma) = \|c_k h_k\|^2_2 + \mu \|H(c_k) - I\|^2_2 - \gamma \left\{ \text{trace}(H(c_k)) - \sqrt{(n - m - 1)\text{trace}(H(c_k)^2)} - \alpha \right\}$$

$$= c_k^2 h_k^2 + \mu (\text{trace}(A^2) + 2c_k h_k \text{trace}(AD) + (c_k)^2 h_k^2 \text{trace}(D^2)) - 2\text{trace}(A) - 2c_k h_k \text{trace}(D) + \text{trace}(I))$$

$$- \gamma (\text{trace}(A) + c_k h_k \text{trace}(D) - \sqrt{(n - m - 1)T(c_k) - \alpha})$$

where,

$$T(c) = \sqrt{\text{trace}(A) + 2c_k h_k \text{trace}(AD) + (c_k)^2 h_k^2 \text{trace}(D^2)}.$$

The system of first-order necessary conditions for problem (5.16) is

$$\nabla_{c_k} \ell(c_k, \gamma) = 2c_k h_k^2 + \mu (2h_k \text{trace}(AD) + 2c_k h_k^2 \text{trace}(D^2) - 2h_k \text{trace}(D))$$

$$- \gamma \left\{ h_k \text{trace}(D) - \frac{(n - m - 1)W(c_k)}{2\sqrt{(n - m - 1)T(c_k)}} \right\}$$

(5.17)

$$g(H(c_k)) = \text{trace}(A) + c_k h_k \text{trace}(D) - \sqrt{(n - m - 1)T(c_k)} \geq \alpha$$

(5.18)

$$\gamma (g(H(c_k)) - \alpha) = 0$$

(5.19)

$$\gamma \geq 0$$

(5.20)

where,

$$W(c_k) = 2h_k \text{trace}(AD) + 2c_k h_k^2 \text{trace}(D^2).$$
The first-order necessary conditions will be satisfied if one of the following cases occurs.

**Case 1:**
If $\gamma = 0$, then (5.19) and (5.20) are satisfied and thus,

$$c_k = \frac{\mu \{\text{trace}(D) - \text{trace}(AD)\}}{h_k + \mu h_k \text{trace}(D^2)}$$ (5.21)

is a solution of the system of first-order necessary conditions ((5.17), (5.18), (5.19) and (5.20)) if $g(H(c_k)) \geq \alpha$.

**Case 2:**
If $\gamma > 0$, then by the complementarity equation (5.19) we have that any $c_k$ satisfying

$$g(H(c_k)) - \alpha = 0 \quad \text{and such that}, \quad \gamma \geq 0$$ (5.22) (5.23)

is a solution of the system of first order necessary conditions for problem (5.16) where,

$$\gamma = \frac{\{2c_k h_k^2 + \mu \{2h_k \text{trace}(AD) + 2(c_k^2 h_k^2 \text{trace}(D^2) - 2h_k \text{trace}(D))\}\}}{2T(c_k) \ h_k \text{trace}(D) - \sqrt{(n - m - 1)} \ W(c_k)}.$$ 

In order to obtain the zeros of the equation (5.22) we need the following result.

**Lemma 5.4** Let

$$\tilde{g}(H(c)) = (\text{trace}(A) + ch_k \text{trace}(D) - \alpha)^2 - (n - m - 1) \ T(c)^2.$$ 

Then,

$$\{c \in \mathbb{R} : g(H(c)) - \alpha = 0\} \subseteq \{c \in \mathbb{R} : \tilde{g}(H(c)) = 0\}.$$ 

**Proof** We can write $g(H(c)) - \alpha$ as

$$g(H(c)) - \alpha = \frac{V_1(c)V_2(c)}{V_2(c)}$$ (5.24)

where,

$$V_1(c) = (\text{trace}(A) + ch_k \text{trace}(D) - \alpha) - \sqrt{(n - m - 1)} \ T(c)$$

$$V_2(c) = (\text{trace}(A) + ch_k \text{trace}(D) - \alpha) + \sqrt{(n - m - 1)} \ T(c)$$
If \( c \) satisfies \( g(H(c)) - \alpha = 0 \) then
\[
V_1(c) V_2(c) = \hat{g}(H(c)) = 0 \quad \text{and},
\]
\[
V_2(c) \neq 0.
\]
(5.25)

As a consequence of this result we have that it is sufficient, for computing the zeros of \( g - \alpha \), to compute the zeros of \( \hat{g} \), a polynomial of degree two in \( \mathbb{R} \). To verify if \( c \) is a zero of \( g - \alpha \) we just compute the two zeros of \( \hat{g} \) and check if equation (5.25) is satisfied.

### 5.3 Numerical Results for \( m=1 \)

In this section we present some preliminary numerical results for the case \( m = 1 \). Since one of the more popular Lagrange multiplier approximation formulas is the least-squares multiplier formula (3.4) we present the numerical results using the least-squares multiplier formula (3.4). That is, our choice of the multiplier will be
\[
\lambda_k = U_{LS}(x_k) + c_k h(x_k)
\]
(5.26)

where \( c_k \) is obtained by Algorithm 5.5 and,
\[
U_{LS}(x) = - (\nabla h(x)^T \nabla h(x))^{-1} \nabla h(x)^T \nabla f(x).
\]
(5.27)

Therefore, the SQP-Newton method we implement to test our multiplier estimate (5.26) is
\[
x_{k+1} = x_k + \Delta x_k
\]
(5.28)

where \( \Delta x_k \) is the solution of the linear system
\[
\begin{pmatrix}
\nabla_x^2 \ell(x_k, U_{LS}(x_k)) + c_k h(x_k) & \nabla h(x_k) \\
\nabla h(x_k)^T & 0
\end{pmatrix}
\begin{pmatrix}
\Delta x \\
\Delta \lambda
\end{pmatrix}
= \begin{pmatrix}
-\nabla_x \ell(x_k, U_{LS}(x_k)) \\
h(x_k)
\end{pmatrix}
\]
(5.29)

and \( c_{k+1} \) is computed using the previous algorithm.

In the numerical implementation of Algorithm 5.5 we imposed the condition
\[
\|c_{k+1}\|_2 \leq M,
\]
(5.30)

for some fixed positive constant \( M \). We proceed in the following way: whenever \( c_{k+1} \) satisfies (5.30), it is acceptable. Otherwise we set \( c_{k+1} = 0 \), and perform the modified
Cholesky factorization, given in Dennis and Schnabel [5], on the matrix $H(0)$, to obtain $p_{k+1}$.

In order to study the robustness of the Lagrange multiplier (5.26) in the SQP-Newton framework we tested each problem starting from various initial points $x_0$. Moreover, in order to make more uniform comparisons, in all the experiments we use the same initial Lagrange multiplier

$$\lambda^0_k = U_{LS}(x_0).$$

(5.31)

The problems we tested were taken from Hock and Schittkowski [10] and the SQP-Newton method, given by (5.28), was implemented using Matlab 4.0. The results are presented in Tables 5.1, 5.2, 5.3 and 5.4. The numbers in the column labeled $PN$ give the number of the problem being tested (as they appear in Hock and Schittkowski [10]). The numbers in the column labeled $NIP$ give the number of starting points used for each particular problem. We choose the initial guess that appears in [10] and several other initial guesses that were presented in Williamson [29]. The numbers in the column under the label $\lambda(x_k) = U(x_k) + c_k h(x_k)$, give the number of different starting points for which the algorithm converged for particular choice rules for $(c_k, \rho_k)$. Observe that the column labeled $c = 0, \rho = 0$ gives the number of starting points for which the SQP-Newton method with the least-squares multiplier as the Lagrange multiplier converges. In this case, $\rho = 0$ means that we did not add anything to the diagonal of the matrix $H(c_k)$ even when the matrix $H(c_k)$ is not positive definite. Moreover, $\rho = \rho_k$ means that we added to the diagonal of $H(c_k)$ the matrix $\rho_k I$. The word "no" or "yes" in the column labeled $H(\lambda_k)$ signifies that the reduced Hessian matrix of the Lagrangian function was not positive definite or was positive definite respectively. The column labeled $\text{cond}(H(\lambda_k))$ gives the condition number of the reduced Hessian matrix. The numbers in the column labeled $\lambda_k$ give the value of the Lagrange multiplier at iteration $k$.

In Tables 5.1, 5.3 and 5.4 the stopping criteria employed was

$$\|(\nabla \ell(x_k, U_{LS}(x_k)), h(x_k))\|_2 \leq 10^{-7},$$

or the number of iterations reached 250. This means that when the reduced Hessian matrix is not positive definite and problem (3.13) may therefore not have a solution, we obtain the iterate instead by solving the extended system (5.29). In Table 5.2 we consider that the algorithm had failed if the reduced Hessian matrix of the Lagrangian
function $H(c_k)$ given in (5.1) has an eigenvalue less than $10^{-6}$ or if the iteration sequence diverges, i.e., the number of iterations reached 250.

We observe from Tables 5.1 and 5.2 that we can achieve convergence to $x_*$ in almost all the problems for different initial iterates $x_0$ using (5.26) the Lagrange multiplier estimate proposed in this study. However, the SQP-Newton method using as Lagrange multiplier the least-squares estimate (5.27) does not converge in many of the problems we tested. In Tables 5.3 and 5.4 we observe that with the choice of Lagrange multiplier (5.26) it was possible to make the reduced Hessian matrix of the Lagrangian function positive definite in such a way that convergence was achieved even when the condition number of the reduced Hessian matrix was not improved. However, the condition number was improved in many cases. It is also important to observe in Tables 5.3 and 5.4 that the reduced Hessian matrix of the Lagrangian function was never positive definite when the least-squares multiplier (5.27) was used as the Lagrange multiplier in the SQP-Newton method. This behavior is demonstrated in almost all the problems we tested. Tables 5.1 and 5.2 demonstrate that the method is not sensitive to the value of $\mu$. We believe, this is because we first check to see if $c_k = 0$ is acceptable. Hence we need not force $\mu \to 0$ to have $c_k \to 0$. Moreover, under the standard assumptions, we will choose $c_k = 0$ near a solution. Later, in Chapter 9 we will present more numerical results for the case $m \geq 1$, not only using the least-squares approximation formula (5.27) but also using the Miele-Cragg-Levy multiplier estimate (3.5) and the QP multiplier estimate (3.7) in the proposed Lagrange multiplier formula (5.26).
\( \lambda_k = U_{LS}(x_k) + c_k h(x_k) \)

<table>
<thead>
<tr>
<th>PN</th>
<th>NIP</th>
<th>( c = 0 )</th>
<th>( c = c_k )</th>
<th>( c = c_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( \rho = 0 )</td>
<td>( \rho = 0 )</td>
<td>( \mu = 500 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( \rho = 0 )</td>
<td>( \mu = 1000 )</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>2</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>26</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>27</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

**Table 5.1** Number of starting points for which the SQP-Newton method converges, (indefinite reduced hessian allowed).

\( \lambda_k = U_{LS}(x_k) + c_k h(x_k) \)

<table>
<thead>
<tr>
<th>PN</th>
<th>NIP</th>
<th>( c = 0 )</th>
<th>( c = c_k )</th>
<th>( c = c_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( \rho = 0 )</td>
<td>( \rho = 0 )</td>
<td>( \mu = 500 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( \rho = 0 )</td>
<td>( \mu = 1000 )</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>2</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>26</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>27</td>
<td>7</td>
<td>1</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

**Table 5.2** Number of starting points for which the SQP-Newton method converges, (indefinite reduced Hessian not allowed).
### Table 5.3  SQP-Newton method for problem #60.  
*converges to (3.38, -0.993, 1.108) which is not a local minimizer.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>$\lambda_k$</th>
<th>$H(\lambda_k)$</th>
<th>$\text{cond}(H(\lambda_k))$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-8.69</td>
<td>no</td>
<td>33.06</td>
</tr>
<tr>
<td>1</td>
<td>-7.12</td>
<td>no</td>
<td>18.63</td>
</tr>
<tr>
<td>2</td>
<td>-6.74</td>
<td>no</td>
<td>1.55</td>
</tr>
<tr>
<td>3</td>
<td>-7.11</td>
<td>no</td>
<td>5.75</td>
</tr>
<tr>
<td>4</td>
<td>-7149</td>
<td>no</td>
<td>7.77</td>
</tr>
<tr>
<td>5</td>
<td>-6.94</td>
<td>no</td>
<td>7.65</td>
</tr>
<tr>
<td>6</td>
<td>-6.82</td>
<td>no</td>
<td>7.23</td>
</tr>
<tr>
<td>7</td>
<td>-6.817</td>
<td>no</td>
<td>7.16</td>
</tr>
<tr>
<td>8</td>
<td>-6.817</td>
<td>no</td>
<td>7.16</td>
</tr>
<tr>
<td>9</td>
<td>-6.817</td>
<td>no</td>
<td>7.16 *</td>
</tr>
</tbody>
</table>

### Table 5.4  SQP-Newton method for problem #60.  
† converges to (1.1, 1.19, 1.53) which is a local minimizer.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>$c_k$</th>
<th>$\lambda_k$</th>
<th>$H(\lambda_k)$</th>
<th>$\text{cond}(H(\lambda_k))$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9.59e+03</td>
<td>-1.53</td>
<td>yes</td>
<td>1622.64</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-0.988</td>
<td>yes</td>
<td>127.20</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-0.721</td>
<td>yes</td>
<td>51.07</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-0.491</td>
<td>yes</td>
<td>23.16</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>-3.17e-01</td>
<td>yes</td>
<td>10.74</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>-0.181</td>
<td>yes</td>
<td>5.24</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>-8.03e-02</td>
<td>yes</td>
<td>2.89</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>-2.88e-02</td>
<td>yes</td>
<td>2.25</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>-1.36e-02</td>
<td>yes</td>
<td>2.52</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>-1.09e-02</td>
<td>yes</td>
<td>2.74</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>-1.072e-02</td>
<td>yes</td>
<td>2.76</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>-1.072e-02</td>
<td>yes</td>
<td>2.76 †</td>
</tr>
</tbody>
</table>
Chapter 6

The $m > 1$ Constraint Case

Motivated by the encouraging results obtained for the case $m = 1$, we extended Algorithm 5.5 to the case where the number of constraints $m$ is greater than one. We present two distinct algorithms for determining a penalty vector $c_k$ in the Lagrange multiplier formula (4.5) keeping objectives (C1), (C2) and (C3) in mind.

In order to introduce the first algorithm we write the expression of the reduced Hessian matrix of the Lagrangian function $H(c)$ when the multiplier is given by (4.5) as:

$$H(c) = B(x_k)^T \nabla_x^2 f(x_k) + C^T h(x_k)B(x_k)$$
$$= B(x_k)^T \left( \nabla_x^2 f(x_k) + \sum_{i=1}^{m} U_i(x_k) \nabla_x^2 h_i(x_k) \right) B(x_k) +$$
$$\sum_{i=1}^{m} c_i h_i(x_k)B(x_k)^T \nabla_x^2 h_i(x_k)B(x_k). \quad (6.1)$$

As in (5.1), we have that, for any given iterate $k$, all the elements in (6.1) are known except the value of the penalty vector $c$. Therefore,

$$H(c) = A + \sum_{i=1}^{m} U_i(x_k)B(x_k)^T \nabla_x^2 h_i(x_k)B(x_k) + \sum_{i=1}^{m} c_i h_i(x_k) \ B(x_k)^T \nabla_x^2 h_i(x_k)B(x_k)$$
$$= A + \sum_{i=1}^{m} U_i^i D^i + \sum_{i=1}^{m} c_i h_i D^i, \quad (6.2)$$

where,

$$h_i = h_i(x_k)$$
$$U_i^i = U_i(x_k)$$
$$A = B(x_k)^T \nabla_x^2 f(x_k)B(x_k)$$
$$D^i = B(x_k)^T \nabla_x^2 h_i(x_k)B(x_k).$$
On the other hand, we can write (6.2) in the following way,

\[ H(c) = A + \sum_{i=1}^{m} U_k^i D^i + \sum_{i=1}^{m} c_i h_i D^i \]
\[ = \frac{1}{m} \left\{ mA + m \sum_{i=1}^{m} U_k^i D^i + m \sum_{i=1}^{m} c_i h_i D^i \right\} \]
\[ = \frac{1}{m} \left\{ (A_1 + mc_1 h_1 D^1) + (A_2 + mc_2 h_2 D^2) + \ldots + (A_m + mc_m h_m D^m) \right\} \]
\[ = \frac{1}{m} \left\{ H_1(c_1) + H_2(c_2) + \ldots + H_m(c_m) \right\} \] \quad (6.3)

where,
\[ A_i = A + mU_k^i D^i \quad \text{and} \]
\[ H_i(c) = A_i + mch_i D^i \quad \text{for} \quad i = 1, \ldots, m. \]

Since the representation of each matrix \( H_i \), for \( i = 1, \ldots, m \), in (6.3), only depends on one penalty parameter \( c_i \), we define the set \( \Omega \) for the case \( m \geq 1 \) as
\[ \Omega = \{ c \in \mathbb{R}^m : g(H(c)) \geq \alpha \}, \] \quad (6.4)

where \( H(c) = A + \sum_{i=1}^{m} U_k^i D^i + \sum_{i=1}^{m} c_i h_i D^i \), and we also define for \( i = 1, \ldots, m \)
\[ \Omega_i = \{ c \in \mathbb{R}^m : g(H_i(c)) \geq \alpha \} \]
\[ \Omega_i = \{ c \in \mathbb{R}^m : g(H_i(c)) \geq \alpha \} \] \quad (6.5)

where \( H_i(c) = A_i + mch_i D^i \). Now, we can state an extension of Algorithm 5.5, for computing the vector \( (c_k, \rho_k) \), when \( m > 1 \).
Algorithm 6.1

Given $\mu$ do the following:

If $\dot{H}(0,0) \in \Omega$
Set $(c_k, \rho_k) = (0, 0)$

Else

For $i = 1, \ldots m$ solve the following constrained problems for $c_i$

\[
\text{(Const)}_i \equiv \min \phi_i(c_i) = \|c_i h_i\|^2 + \mu \|H_i(c_i) - I\|^2 \quad \text{subject to } c_i \in \Omega_i
\]

If $\dot{H}(c_i, 0) \in \Omega_i$
Set $c_i^* = c_i$

Else
Let $c_i^*$ be the solution of unconstrained problem

\[
\text{(Unconst)}_i \equiv \min \phi_i(c_i^*) = \|c_i h_i\|^2 + \mu \|H_i(c_i) - I\|^2
\]

End

End

Set $c^* = (c_1^*, c_m^*, \ldots, c_m^*)$

If $\dot{H}(c^*, 0) \in \Omega$
Set $(c_k, \rho_k) = (c^*, 0)$

Else

Set $(c_k, \rho_k) = (0, \rho_k)$ where $\rho_k$ is computed such that

$\dot{H}(0, \rho_k) = H(0) + \rho_k I \in \text{SPD}$

End

End

Notice that this algorithm is actually a parallel version of Algorithm 5.5, given in Chapter 5, since it is possible to solve each problem \text{(Const)}_i for $i = 1, \ldots m$ independently.

In order to state another version of Algorithm 5.5 we write the reduced Hessian matrix of the Lagrangian function, at the iteration $k$, of the SQP-Newton method as:

\[
H(c) = A + \sum_{i=1}^{m} U_k^i D^i + \sum_{i=1}^{m} c_i h_i D_i
\]

\[
= \left( \frac{1}{m} A + \frac{1}{m} A + \ldots + \frac{1}{m} A \right) + \sum_{i=1}^{m} U_k^i D^i + \sum_{i=1}^{m} c_i h_i D_i
\]  

(6.6)
\[
\begin{align*}
\ &= A_0 + \left( \frac{1}{m} A + \frac{1}{m} A + \ldots + \frac{1}{m} A \right) + \sum_{i=2}^{m} U_i D^i + \sum_{i=1}^{m} c_i h_i D^i \\
\ &= A_{m-2} + \frac{1}{m} A + U_k^m D^m + \sum_{i=m-1}^{m} c_i h_i D^i \\
\ &= A_{m-1} + c_m h_m D^m 
\end{align*}
\]

where,

\[
\begin{align*}
A_0 &= \frac{1}{m} A + U_k D^1 \\
A_i &= \frac{1}{m} A + A_{i-1} + U_k^{i+1} D^{i+1} + c_i h_i D^i \quad i = 1, \ldots, m-1
\end{align*}
\]

Let us denote, for \( i = 1, \ldots, m \)

\[
\begin{align*}
H_i(c) &= A_{i-1} + c h_i D^i \quad \text{and} \\
\hat{\Omega}_i &= \{ c \in \mathbb{R}^m : g(H_i(c)) \geq \frac{i}{m} \alpha \},
\end{align*}
\]

where \( H_i(c) = A_{i-1} + c h_i D^i \).

We will use the expression (6.7) for \( H(c_k) \) and the notation (6.4), (6.8) and (6.9) to present another extension of Algorithm 5.5 when \( m > 1 \).
Algorithm 6.2

For a given value of the constant $\mu$ do the following:

If $\hat{H}(0, 0) \in \Omega$

Set $(c_k, \rho_k) = (0, 0)$

Else

For $i = 1, \ldots, m$ solve the constrained problems for $c_i$

$(\text{Const})_i \equiv \min_{c_i \in \hat{\Omega}_i} \phi_i(c_i) = \|c_i h_i\|_2^2 + \mu \|H_i(c_i) - \frac{i}{m} I\|_F^2$

If $\hat{H}_i(c_i, 0) \in \hat{\Omega}_i$

Set $c_i^* = c_i$

Else

Let $c_i^*$ be the solution of unconstrained problem

$(\text{Unconst})_i \equiv \min_{c_i} \phi_i(c_i) = \|c_i h_i\|_2^2 + \mu \|H_i(c_i) - \frac{i}{m} I\|_F^2$

End

Set $A_i = \frac{1}{m} A + A_{i-1} + (c_i^*)^* h_i B_i$

End

Set $c^* = (c_1^*, c_2^*, \ldots, c_m^*)$

If $\hat{H}(c^*, 0) \in \Omega$

Set $(c_k, \rho_k) = (c^*, 0)$

Else

Set $(c_k, \rho_k) = (0, \rho_k)$ where $\rho_k$ is computed such that

$\hat{H}(0, \rho_k) = H(0) + \rho_k I \in SPD$

End

End

Algorithm 6.2 is a sequential version of Algorithm 5.5, since the solution of problem $(\text{Const})_{i+1}$ depends on the solution of problem $(\text{Const})_i$ for $i = 1, \ldots, m$.

Extensive numerical results, using these two algorithms for computing the vector $(c_k, \rho_k)$, will be presented in Chapter 8.
Chapter 7

Local and q-Quadratic convergence

In this chapter we study the local convergence properties of the SQP-Newton method with the Lagrange multiplier estimate given by formula (4.5). We present our convergence analysis in terms of a generic choice for the penalty parameter \( c \) in formula (4.5). Towards this end let us begin with the following definition.

**Definition 7.1** let \( x_* \) be a stationary point of problem (2.1) and consider the penalty choice function \( c : \mathbb{R}^n \rightarrow \mathbb{R}^m \). We say that \( c \) is locally bounded at \( x_* \) if there exists \( N(x_*) \), a neighborhood of \( x_* \), such that \( c(N(x_*)) \) is a bounded subset of \( \mathbb{R}^m \).

In addition to the standard Newton's method assumptions (A1)-(A3), our convergence theory will require the boundedness assumption:

**A4** The penalty choice function \( c \) is locally bounded at \( x_* \).

We demonstrate local convergence and quadratic convergence for the SQP-Newton method using the Lagrange multiplier choice (4.5) under the standard assumptions (A1)-(A3) and the boundedness assumption (A4). It is not difficult to see that under the standard assumptions (A1)-(A3) our algorithm for choosing the penalty parameter leads to a choice which is locally bounded. Indeed, under the standard assumptions, near a local solution the choice \( (c, \rho) = 0 \) is both acceptable and will be taken by our algorithm.

First, we present a useful general perturbation lemma that resembles Lemma 11.2.2. in Ortega and Rheinboldt [14]. However, our result is more general in the sense that we consider perturbations to both the Jacobian matrix and the right-hand side in the Newton linear system. To begin with consider the nonlinear system

\[ F(x) = 0, \quad (7.1) \]

where \( F : \mathbb{R}^n \rightarrow \mathbb{R}^n \). Now, consider Newton's method applied to (7.1)

\[ x_+ = x - J(x)^{-1} F(x), \quad (7.2) \]
and a perturbation of this method, say

\[ x_+ = x - (J(x) + \hat{J}(x))^{-1}(F(x) + \hat{F}(x)), \]  

(7.3)

where \( \hat{J} : \mathbb{R}^n \to \mathbb{R}^{n \times n} \) and \( \hat{F} : \mathbb{R}^n \to \mathbb{R}^n \). The standard assumptions for Newton’s method applied to system (7.1) are:

(S1) There exists \( x_* \) such that \( F(x_*) = 0 \).

(S2) The function \( F \) is continuously differentiable in an open convex set \( D \) containing, and the Jacobian matrix \( J \) is Lipschitz continuous in \( D \).

(S3) The matrix \( J(x_*) \) is nonsingular.

**Lemma 7.1** (Perturbation Lemma)

Assume the standard assumptions (S1)-(S3). Assume further that in a neighborhood \( \hat{D} \subset D \) of the local solution \( x_* \).

(i) \( \|\hat{J}(x)\| = O(\|x - x_*\|) \),

(ii) \( \|\hat{F}(x)\| = O(\|x - x_*\|^2) \).

Then, the iterative procedure (7.3) is locally and \( q \)-quadratically convergent to \( x_* \).

**Proof** From assumption (i) and Theorem 3.6 of Stewart [18] we have

\[ x_+ - x_* = x - x_* - (J^{-1}(x) + W(x)J^{-1}(x))(F(x) + \hat{F}(x)) \]

where

\[ \|W(x)\| \leq \frac{\|J^{-1}(x)\hat{J}(x)\|}{1 - \|J^{-1}(x)\hat{J}(x)\|}, \]

for all \( x \) in a neighborhood of \( x_* \). Hence,

\[ \|x_+ - x_*\| \leq \|J^{-1}(x)(F(x_* - F(x_*) - J^{-1}(x))(x_* - x))\|
\]

\[ + \|J^{-1}(x)\|\|\hat{F}(x)\| + \|W(x)\|\|J^{-1}(x)\|\|F(x_* - F(x_*))\|
\]

\[ + \|W(x)\|\|J^{-1}(x)\|\|\hat{F}(x)\|. \quad (7.4) \]

By the standard assumptions (S1)-(S3) and assumption (i) we have that there exists a neighborhood of \( x_* \), such that for all \( x \) in this neighborhood

\[ \|W(x)\| = O(\|x - x_*\|) \]

(7.5)
Moreover, from (7.4), (7.5) and assumptions (i), (ii) we have that for all \( x \) in a neighborhood of \( x_* \)
\[
\|x_+ - x_*\| = O(\|x - x_*\|^2).
\] (7.6)
It follows from (7.6) that the iterative procedure (7.3) is locally and quadratically convergent to \( x_* \).

In order to utilize the perturbation lemma we consider Newton's method on the nonlinear system of equations
\[
\nabla_x L(x, \lambda_*, \hat{\rho}) = 0,
\] (7.7)
where \( \lambda_* \) is the multiplier associated with the solution \( x_* \) of problem (2.1) and \( \hat{\rho} \) is a positive constant such that the matrix \( \nabla_x^2 L(x_*, \lambda_*, \hat{\rho}) \) is positive definite. Hence, our ideal algorithm will be
\[
x_+ = x - \nabla_x^2 L(x, \lambda_*, \hat{\rho})^{-1} \nabla_x L(x, \lambda_*, \hat{\rho}).
\] (7.8)
Recall that the iterative procedure (7.8) is locally and quadratically convergent to \( x_* \).

The SQP-Newton method with our choice of the Lagrange multiplier formula (4.5) can be written
\[
x_+ = x - R(x)^{-1} \{ \nabla f(x) + \nabla h(x)\Lambda(x)(h(x) - \nabla h(x)^T R(x)^{-1} \nabla f(x)) \}
\] (7.9)
\[
= x - (\nabla_x^2 L(x, \lambda_*, \hat{\rho}) + Q(x))^{-1} (\nabla_x L(x, \lambda_*, \hat{\rho}) + E(x))
\]
\[
= x - R(x)^{-1} (\nabla_x L(x, \lambda_*, \hat{\rho}) + E(x)),
\] (7.10)
where
\[
R(x) = \nabla_x^2 L(x, \lambda_*, \hat{\rho}) + Q(x)
\]
\[
Q(x) = \nabla_x^2 \ell(x, U(x)) - \nabla_x^2 \ell(x, \lambda_*) + C h(x) \nabla^2 h(x) - \hat{\rho} h(x) \nabla^2 h(x)
\]
\[
\Lambda(x) = (\nabla h(x)^T R(x)^{-1} \nabla h(x))^{-1}
\]
\[
E(x) = \nabla h(x) \{ \Lambda(x)(h(x) - \nabla h(x)^T R(x)^{-1} \nabla f(x)) - \lambda_* - \hat{\rho} h(x) \}.
\]

Hence, the iterative procedure (7.10) has been written as a perturbation of the ideal Newton method given by (7.8), where the perturbations are given by \( Q(x) \) and \( E(x) \).

Our next result demonstrates that the SQP-Newton method with Lagrange multiplier estimate given by formula (4.5) satisfies conditions (i) and (ii) of the previous
lemma, whenever the approximation formula \( U \) in the Lagrange multiplier (4.5) satisfies a mild condition.

**Lemma 7.2** Let \( x_* \) be a local solution of problem (2.1) with associated multiplier \( \lambda_* \). Assume that in a neighborhood of the solution \( x_* \)

\[
\|U(x) - \lambda_*\| = O(\|x - x_*\|). 
\]

(7.11)

Also assume the standard conditions (A1)-(A3) and the boundedness condition (A4). Then there exists a neighborhood \( N(x_*) \), of a local solution \( x_* \), such that for all \( x \in N(x_*) \)

(a) \( \|Q(x)\| = O(\|x - x_*\|) \),

(b) \( \|E(x)\| = O(\|x - x_*\|^2) \).

**Proof** From the boundedness condition (A4), the standard assumption (A1) and the condition (7.11) it follows that there exists a neighborhood of the solution \( x_* \), such that for all \( x \) in that neighborhood

\[
\|Q(x)\| \leq \|\nabla^2_x \ell(x, U(x)) - \nabla^2_x \ell(x_*, \lambda_*)\| + \|c\|\|h(x) - h(x_*)\|\|\nabla^2 h(x)\|
\]

\[
+ \|\hat{\lambda}\|\|h(x) - h(x_*)\|\|\nabla^2 h(x)\| = O(\|x - x_*\|). 
\]

(7.12)

Therefore condition (a) holds. For the sake of simplicity we will denote the \( i^{th} \) component of the gradient vector of \( h_i \), evaluated at \( x_* \), \( (\nabla h_i(x_*)) \) by \( \nabla h_i \). Similar notation will be used for similar quantities. Using this notation we have

\[
\Lambda \{ h - \nabla h^T R^{-1} \nabla f \} - \lambda_* - \hat{\lambda} h =
\]

\[
\Lambda \{ h - \nabla h^T R^{-1} \nabla f - A^{-1} \lambda_* - \hat{\lambda} A^{-1} h \}
\]

\[
= \Lambda \{ (I_{m \times m} - \hat{\lambda} A^{-1}) (h - h(x_*)) - \nabla h^T R^{-1} (\nabla f + \nabla h \lambda_*) \}
\]

\[
= \Lambda \{ P (h - h(x_*)) - \nabla h^T R^{-1} (\nabla f + \nabla h \lambda_*)
\]

\[
+ \nabla h^T R^{-1} (\nabla f(x_*) + \nabla h(x_*) \lambda_*) \}
\]

\[
= \Lambda \left\{ P \int_0^1 \nabla h^T (x_* + t(x - x_*))(x - x_*) dt
\]

\[
- \nabla h^T R^{-1} \int_0^1 \nabla^2 f(x_* + t(x - x_*))(x - x_*) dt
\]

\[
- \nabla h^T R^{-1} \sum_{i=1}^m \int_0^1 \nabla^2 h_i (x_* + t(x - x_*))(x - x_*)(\lambda_*)_i dt \right\}
\]
where

\[ P = I_{m \times m} - \hat{\rho} \Lambda^{-1}. \]

Adding and subtracting some terms conveniently we obtain

\[ \Lambda \{ h - \nabla h^T R^{-1} \nabla f \} - \lambda - \hat{\rho} h_k = \]

\[ \Lambda \left\{ P \left( \int_0^1 (\nabla h(x_* + t(x-x_*)) - \nabla h(x_*))^T (x-x_*) dt \right) + \nabla h(x_*)^T (x-x_*) \right. \]

\[ - \nabla h^T R^{-1} \left( \int_0^1 (\nabla^2 f(x_* + t(x-x_*)) - \nabla^2 f(x_*))(x-x_*) dt \right) + \nabla^2 f(x_*)(x-x_*) \right) \]

\[ - \nabla h^T R^{-1} \left( \sum_{i=1}^m \int_0^1 (\nabla^2 h_i(x_* + t(x-x_*)) - \nabla^2 h_i(x_*))(x-x_*)(\lambda_*)_i dt \right) \]

\[ + \sum_{i=1}^m \nabla^2 h_i(x_*)(x-x_*)(\lambda_*)_i \right\}. \]  \tag{7.13}

We can rewrite (7.13) as

\[ \Lambda \{ h - \nabla h^T R^{-1} \nabla f \} - \lambda - \hat{\rho} h_k = \]

\[ \Lambda \left\{ P \int_0^1 A^1(t)(x-x_*) dt - \nabla h^T R^{-1} \int_0^1 A^2(t)(x-x_*) dt \right. \]

\[ - \nabla h^T R^{-1} \sum_{i=1}^m \int_0^1 A^3_i(t)(\lambda_*)_i(x-x_*) dt \right\} + \Lambda Z(x-x_*) \]  \tag{7.14}

where

\[ A^1(t) = (\nabla h(x_* + t(x-x_*)) - \nabla h(x_*))^T, \]

\[ A^2(t) = \nabla^2 f(x_* + t(x-x_*)) - \nabla^2 f(x_*), \]

\[ A^3_i(t) = \nabla^2 h_i(x_* + t(x-x_*)) - \nabla^2 h_i(x_*), \quad \text{and} \]

\[ Z = P \nabla h(x_*)^T - \nabla h^T R^{-1} \nabla^2 f(x_*) - \nabla h^T R^{-1} \sum_{i=1}^m \nabla^2 h_i(x_*)(\lambda_*)_i. \]  \tag{7.15}

By assumption (A1) and the fact that 0 ≤ t ≤ 1 we have that in a neighborhood of the solution x_*

\[ \|A^1(t)\| = O(\|x-x_*\|), \]

\[ \|A^2(t)\| = O(\|x-x_*\|), \]

\[ \|A^3_i(t)\| = O(\|x-x_*\|). \]  \tag{7.16}
On the other hand, adding and subtracting some terms to (7.15) we have

\[
Z = P\nabla h(x_*)^T - \nabla h R^{-1}(\nabla^2 f(x_*) + \sum_{i=1}^m \nabla^2 h_i(x_*)(\lambda_*)_i)
\]
\[
= (I - \hat{\rho}\nabla h R^{-1}\nabla h)\nabla h(x_*)^T - \nabla h^T R^{-1}(\nabla^2 f(x_*) + \sum_{i=1}^m \nabla^2 h_i(x_*)(\lambda_*)_i)
\]
\[
= \nabla h(x_*)^T R^{-1} R - \hat{\rho}\nabla h^T R^{-1} \nabla h \nabla h(x_*)^T - \nabla h^T R^{-1}(\nabla^2 f(x_*) + \sum_{i=1}^m \nabla^2 h_i(x_*)(\lambda_*)_i)
\]
\[
= \nabla h^T R^{-1} \{(\nabla^2 f - \nabla^2 f(x_*)) \nabla h - \nabla h(x_*)\} + \sum_{i=1}^m \nabla^2 h_i(U_i - (\lambda_*)_i)
\]
\[
+ \hat{\rho}\nabla h(\nabla h - \nabla h(x_*)) + \sum_{i=1}^m \nabla^2 h_i(U_i - (\lambda_*)_i)
\]
\[
+ (\nabla h(x_*) - \nabla h)^T R^{-1} \nabla^2 f + \hat{\rho} R^{-1} \nabla h \nabla h^T
\]
\[
+ R^{-1} \sum_{i=1}^m \nabla^2 h_i U_i + \nabla h(x_*)^T R^{-1} \sum_{i=1}^m c_i \nabla^2 h_i (h_i - h_i(x_*)).
\]

(7.17)

By assumptions (A1)-(A4) and condition (7.11) we have that in a neighborhood of the solution $x_*$

\[
\|Z\| = O(\|x - x_*\|)
\]

(7.18)

From (7.14), (7.16) and (7.18) it follows that there exists a neighborhood of the solution $x_*$ such that

\[
\|E(x)\| \leq \|\nabla h\| \|A\{h - \nabla h^T R^{-1} \nabla f\} - \lambda_* - \hat{\rho} h\|
\]
\[
= O(\|x - x_*\|^2).
\]

\[
\square
\]

**Theorem 7.1** Let $x_*$ be a local solution of problem (2.1). Assume the standard conditions (A1)-(A3) and the boundedness condition (A4). Also assume that in a neighborhood of the solution $x_*$

\[
\|U(x) - \lambda_*\| = O(\|x - x_*\|).
\]

(7.19)

Then the SQP-Newton method with the choice of Lagrange multiplier given by (4.5) is locally and $q$-quadratically convergent to $x_*$. 
Proof It is not difficult to prove and well known that the standard assumptions (A1)-(A3) imply the standard assumptions (S1)-(S3) for the nonlinear system of equations (7.7). On the other hand, by Lemma 7.2 conditions (ii) and (iii) of Lemma 7.1 are satisfied. Therefore, we have the assumptions of Lemma 7.1 and thus the iterative procedure (7.10) is locally and q-quadratically convergent to \( x_* \).

\[ \square \]

Corollary 7.1 Let \( x_* \) be a local solution of problem (2.1) with associated Lagrange multiplier \( \lambda_* \). The SQP-Newton method with Lagrange multiplier estimate given by (4.5), where the approximation formula \( U \) is the least-squares formula (3.4) or the Micle-Cragg-Levy formula (3.5) with the penalty parameter \( c \) chosen accordingly to Algorithm 6.1 and Algorithm 6.2, is locally and q-quadratically convergent to \( x_* \).

Proof The penalty vector \( c \) obtained by Algorithm 6.1 and Algorithm 6.2 satisfies the boundedness condition (A4). It is straightforward to prove that under the standard assumptions (A1)-(A3), there exists a neighborhood of the local solution \( x_* \) such that

\[ \|U(x) - \lambda_*\| = O(\|x - x_*\|) \]  

where the approximation formula \( U \) is given by formula (3.4) or by formula (3.5).

\[ \square \]

A rather direct extension of these results shows that the SQP-Newton method with Lagrange multiplier formula given by (4.5), where formula \( U \) is given by the QP multiplier formula (3.7) is locally and q-quadratically convergent to \( (x_*, \lambda_*) \).
Chapter 8

Numerical Results

In this chapter we discuss some issues concerning the implementation of the SQP-Newton method and present numerical results obtained from our implementation of the method.

8.1 Implementational Issues

The SQP-Newton method that we implemented to test our multiplier estimate is

\[ x_{k+1} = x_k + \Delta x_k \]  
(8.1)

where \( \Delta x_k \) is the solution of the linear system

\[
\begin{pmatrix}
\nabla^2 \ell(x_k, U(x_k)) + C_k h(x_k) & \nabla h(x_k) \\
\n\nabla h(x_k)^T & 0 \\
\end{pmatrix}
\begin{pmatrix}
\Delta x \\
\Delta \lambda \\
\end{pmatrix}
= 
\begin{pmatrix}
-\nabla_x \ell(x_k, U(x_k)) \\
-h(x_k) \\
\end{pmatrix}.
\]  
(8.2)

The penalty vector \( C_{k+1} = \text{diag}(c_{k+1}) \) is computed using Algorithm 6.1 or Algorithm 6.2. We also imposed the condition

\[ \|c_{k+1}\|_2 \leq M, \]  
(8.3)

for some positive constant \( M \). We proceed in the following way: whenever \( c_{k+1} \) satisfies (8.3), it is acceptable. Otherwise, we set \( c_{k+1} = 0 \), and perform the modified Cholesky factorization, given in Dennis ans Schnabel [5], on the matrix \( H(0) \) to obtain \( \rho_{k+1} \).

In order to study the robustness of the Lagrange multiplier choice (4.5) in the SQP-Newton framework we tested each problem starting from various initial points \( x_0 \). Moreover, in order to make uniform comparisons, in all the experiments we use the same initial Lagrange multiplier

\[ \lambda^0_k = U_{L,S}(x_0). \]  
(8.4)
8.2  Numerical Results

The problems tested were taken from Hock and Schittkowski [10] and will be referenced by the number given there. The SQP-Newton method with the choice of the multiplier (4.5) was implemented in Matlab 4.0. The choices for $U$ in the formula (4.5) are:

\begin{align*}
U_{LS}(x) &= - (\nabla h(x)^T \nabla h(x))^{-1} \nabla h(x)^T \nabla f(x) \\
U_{MCL}(x) &= (\nabla h(x)^T \nabla h(x))^{-1} (h(x) - \nabla h(x)^T \nabla f(x)) \\
U_{QP}(x, \lambda) &= (\nabla h(x)^T H \nabla h(x))^{-1} (h(x) - \nabla h(x)^T H \nabla f(x))
\end{align*}

where the matrix $H = \nabla^2_{xx} \ell(x, \lambda)^{-1}$.

The numbers in the column labeled $PN$ give the number of the problem being tested. The numbers in the column labeled $NIP$ give the number of starting points tested for each particular problem. We choose the initial guess that appears in Hock and Schittkowski [10] and several other initial guesses that were presented in Williamson [29]. The numbers in the column under the label $\lambda(x_k) = U(x_k) + C_k h(x_k)$, give the number of different starting points for which the algorithm converged. For example, in the column labeled $c = c_k$, $\rho = 0$, $\mu = 500$ and $Seq.$ appears the number of starting points for which the SQP-Newton method, given by (8.1) with $\rho = 0$ and $\mu = 500$ in the sequential version of Algorithm 5.5, converged. In this case $\rho = 0$ means that we did not add anything to the diagonal of the matrix $H(c_k)$ even when the matrix $H(c_k)$ is not positive definite. Moreover, $\rho = \rho_k I$ means that we added to the diagonal of $H(c_k)$ the matrix $\rho_k I$. The numbers in the column labeled $\frac{\mu}{\rho}$ give the number of iterations required to achieve convergence over the number of times the value of $\rho$ was computed.

In Tables 8.1, 8.2, 8.3, 8.7, 8.8 and 8.9 the stopping criteria employed was

$$
\| (\nabla \ell(x_k, U(x_k)), h(x_k)) \|_2 \leq 10^{-7},
$$

or the number of iterations reached 250. This means that when the reduced Hessian matrix is not positive definite and problem (3.13) may therefore not have a solution, we obtain the iterate instead by solving the extended system (8.2). In Tables 8.4, 8.5 and 8.6 we consider that the algorithm had failed if the reduced Hessian matrix of the Lagrangian function $H(c_k)$ given in (6.1) had an eigenvalue less than $10^{-6}$ or if the iteration sequence diverged, i.e., the number of iterations reached 250.

We observe from Tables 8.1, 8.2, 8.3, 8.4, 8.5 and 8.6 that we can achieve convergence to a minimizer $x_*$, in many problems from different starting points, just by
computing the penalty vector $c_k$, with Algorithm 6.1 or 6.2, in the Lagrange multiplier (8.1). Also Tables 8.4, 8.5 and 8.6 indicate that this new choice of the multiplier generates a positive definite reduced Hessian matrix more frequently than the traditional multiplier formulas (least-squares multiplier (8.5), Miele-Cragg-Levy multiplier (8.6) and the QP multiplier (8.7)). Finally from Tables 8.7, 8.8 and 8.9 we observe that even when we cannot find a penalty parameter such that the reduced Hessian matrix is positive definite (in which case we add $\rho_k I$ to $H(0)$) the number of times we compute $\rho_k$ compared with the number of times we compute $\rho_k$ for the traditional multipliers is smaller. Moreover, we can achieve convergence to a minimizer in almost all the problems from most starting points. It is important to mention that the algorithms do not always converge to the same points.
\[
\lambda_k = U_{LS}(x_k) + C_k h(x_k)
\]

<table>
<thead>
<tr>
<th>PN</th>
<th>NIP</th>
<th>(c = 0)</th>
<th>(c = c_k)</th>
<th>(c = c_k)</th>
<th>(c = c_k)</th>
<th>(c = c_k)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(\rho = 0)</td>
<td>(\rho = 0)</td>
<td>(\mu = 500)</td>
<td>(\mu = 500)</td>
<td>(\mu = 1000)</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>26</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>27</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>39</td>
<td>11</td>
<td>0</td>
<td>9</td>
<td>7</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>40</td>
<td>12</td>
<td>4</td>
<td>11</td>
<td>7</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>42</td>
<td>12</td>
<td>4</td>
<td>9</td>
<td>10</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>77</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>78</td>
<td>12</td>
<td>6</td>
<td>12</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>79</td>
<td>10</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>46</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>47</td>
<td>11</td>
<td>5</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>56</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>7</td>
</tr>
</tbody>
</table>

**Table 8.1** Number of starting points for which the SQP-Newton method converges, (indefinite reduced Hessian allowed).
<table>
<thead>
<tr>
<th>$PN$</th>
<th>$NIP$</th>
<th>$c = 0$</th>
<th>$c = c_k$</th>
<th>$c = c_k$</th>
<th>$c = c_k$</th>
<th>$c = c_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\rho = 0$</td>
<td>$\mu = 500$</td>
<td>$\rho = 0$</td>
<td>$\mu = 500$</td>
<td>$\rho = 0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$Parallel$</td>
<td>$Seq.$</td>
<td>$Parallel$</td>
<td>$Seq.$</td>
<td>$Parallel$</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>3</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>26</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>27</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>39</td>
<td>11</td>
<td>5</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>40</td>
<td>12</td>
<td>1</td>
<td>9</td>
<td>7</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>42</td>
<td>12</td>
<td>7</td>
<td>9</td>
<td>10</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>77</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>78</td>
<td>12</td>
<td>7</td>
<td>8</td>
<td>11</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>79</td>
<td>10</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>46</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>47</td>
<td>11</td>
<td>6</td>
<td>11</td>
<td>10</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>56</td>
<td>9</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 8.2 Number of starting point for which the SQP-Newton method converges, (indefinite reduced Hessian allowed).
<table>
<thead>
<tr>
<th>PN</th>
<th>NIP</th>
<th>$\lambda_k = U_{QP}(x_k) + C_k h(x_k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$c = 0$ $\rho = 0$ $\mu = 500$ Parallel</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>27</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>39</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>77</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>78</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>79</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>46</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>47</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>56</td>
<td>9</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 8.3 Number of starting points for which the SQP-Newton method converges, (indefinite reduced Hessian allowed).
<table>
<thead>
<tr>
<th>PN</th>
<th>NIP</th>
<th>( c = 0 )</th>
<th>( c = c_\ast )</th>
<th>( c = c_\ast )</th>
<th>( c = c_\ast )</th>
<th>( c = c_\ast )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( \rho = 0 )</td>
<td>( \mu = 500 )</td>
<td>( \mu = 1000 )</td>
<td>( \rho = 0 )</td>
<td>( \mu = 500 )</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>26</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>27</td>
<td>7</td>
<td>1</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>39</td>
<td>11</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>40</td>
<td>12</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>42</td>
<td>12</td>
<td>3</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>77</td>
<td>10</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>78</td>
<td>12</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>79</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>46</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>47</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>56</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 8.4** Number of starting points for which the SQP-Newton method converges, (indefinite reduced Hessian not allowed).
\begin{table}
\centering
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline
$PN$ & $NIP$ & $c = 0$ & $c = c_\star$ & $c = c_\star$ & $c = c_\star$ & $c = c_\star$ \\
$\rho = 0$ & $\mu = 0$ & $\mu = 1000$ & $\mu = 500$ & $\mu = 1000$ & $\mu = 500$ & $\mu = 1000$ \\
& & Parallel & Seq. & Seq. & Parallel & Seq. \\
\hline
6 & 9 & 3 & 6 & 7 & 7 & 6 \\
7 & 4 & 0 & 4 & 4 & 4 & 4 \\
26 & 5 & 2 & 4 & 4 & 0 & 0 \\
27 & 7 & 5 & 7 & 7 & 7 & 7 \\
60 & 4 & 1 & 4 & 4 & 4 & 4 \\
39 & 11 & 3 & 6 & 6 & 7 & 6 \\
40 & 12 & 3 & 2 & 2 & 2 & 3 \\
42 & 12 & 3 & 9 & 9 & 10 & 10 \\
77 & 10 & 4 & 3 & 3 & 3 & 2 \\
78 & 12 & 6 & 7 & 7 & 7 & 7 \\
79 & 10 & 5 & 4 & 4 & 5 & 5 \\
46 & 8 & 0 & 0 & 0 & 0 & 0 \\
47 & 11 & 2 & 4 & 4 & 5 & 4 \\
56 & 9 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{tabular}
\caption{Number of starting points for which the SQP-Newton method converges, (indefinite reduced Hessian not allowed).}
\end{table}
<table>
<thead>
<tr>
<th>PN</th>
<th>NIP</th>
<th>$\lambda_k = U_{QP}(x_k, \lambda_k) + C_k h(x_k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$c = 0$ $\rho = 0$</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>.6</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>27</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>39</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>77</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>78</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>79</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>46</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>47</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>56</td>
<td>9</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 8.6** Number of starting points for which the SQP-Newton method converges, (indefinite reduced Hessian not allowed).
\[
\lambda_k = U_{LS}(x_k) + C_k h(x_k)
\]

<table>
<thead>
<tr>
<th>(PN)</th>
<th>(NIP)</th>
<th>(c = 0)</th>
<th>(c = c_k)</th>
<th>(c = c_k)</th>
<th>(c = c_k)</th>
<th>(c = c_k)</th>
<th>(c = c_k)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(p = p_k)</td>
<td>(p = p_k)</td>
<td>(p = p_k)</td>
<td>(p = p_k)</td>
<td>(p = p_k)</td>
<td>(p = p_k)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\mu = 500)</td>
<td>(\mu = 1000)</td>
<td>(\mu = 500)</td>
<td>(\mu = 1000)</td>
<td>(\mu = 500)</td>
<td>(\mu = 1000)</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>110</td>
<td>12</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>173</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>59</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>39</td>
<td>11</td>
<td>343</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>40</td>
<td>12</td>
<td>145</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>42</td>
<td>12</td>
<td>184</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>77</td>
<td>10</td>
<td>219</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>78</td>
<td>12</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>79</td>
<td>10</td>
<td>112</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>46</td>
<td>8</td>
<td>228</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>47</td>
<td>11</td>
<td>204</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>56</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 8.7 Number of starting points for which the SQP-Newton method converges.
\[ \lambda_k = U_{MCL}(x_k) + C_k h(x_k) \]

<table>
<thead>
<tr>
<th>PN</th>
<th>NIP</th>
<th>( \frac{c = 0}{\mu} \rho = \rho_k )</th>
<th>( \frac{c = c_k}{\mu} \rho = \rho_k )</th>
<th>Parallel</th>
<th>( \frac{c = c_k}{\mu} \rho = \rho_k )</th>
<th>Seq.</th>
<th>( \frac{c = c_k}{\mu} \rho = \rho_k )</th>
<th>Parallel</th>
<th>Seq.</th>
<th>( \frac{c = c_k}{\mu} \rho = \rho_k )</th>
<th>Seq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>9</td>
<td>9 544 148 9 504 58 9 504 42 9 504 58 9 504 42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>4 96 4 4 54 6 4 54 4 4 54 6 4 54 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>5</td>
<td>5 114 12 5 147 3 5 147 3 5 147 3 5 147 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>4</td>
<td>7 88 3 7 122 6 7 122 6 7 122 6 7 122 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>4 50 5 4 63 6 4 64 6 4 63 6 4 64 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>11</td>
<td>8 142 34 10 225 20 9 164 3 10 215 27 9 146 24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>12</td>
<td>9 114 61 5 46 7 6 64 8 7 108 46 7 85 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>12</td>
<td>12 163 10 12 128 3 12 128 3 12 115 2 12 115 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>10</td>
<td>9 418 245 9 228 238 8 161 15 8 152 23 9 128 29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>12</td>
<td>8 51 6 10 86 5 10 86 5 9 98 36 9 98 36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>10</td>
<td>10 122 20 10 122 16 10 122 16 10 111 13 10 120 12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>8</td>
<td>8 226 53 8 271 43 8 265 42 7 217 40 8 260 57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>11</td>
<td>11 260 58 11 221 13 11 221 13 11 252 61 11 242 27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>9</td>
<td>6 856 756 9 477 260 6 76 35 1 6 1 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8.8 Number of starting points for which the SQP-Newton method converges.
<table>
<thead>
<tr>
<th>PN</th>
<th>NIP</th>
<th>$c = 0$ $\mu = 500$ Parallel</th>
<th>$c = c_k$ $\mu = 500$ Seq.</th>
<th>$c = c_k$ $\mu = 1000$ Parallel</th>
<th>$c = c_k$ $\mu = 1000$ Seq.</th>
<th>$c = c_k$ $\mu = 500$ Seq.</th>
<th>$c = c_k$ $\mu = 1000$ Seq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>9</td>
<td>9 50/32 9 292/2 9 189/2 9 282/2 9 189/2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>4 450/64 4 52/0 4 52/0 4 52/0 4 52/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>5</td>
<td>5 118/16 5 181/0 5 181/0 5 181/0 5 181/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>4</td>
<td>6 81/5 6 139/1 6 139/1 6 139/1 6 139/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>6 69/7 4 59/0 4 59/0 4 59/0 4 59/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>11</td>
<td>10 210/50 10 258/65 10 244/57 9 170/42 9 167/28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>12</td>
<td>9 145/88 9 144/31 9 121/42 9 145/40 9 86/28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>12</td>
<td>9 189/68 9 188/65 7 115/25 6 184/72 8 184/65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>10</td>
<td>8 322/146 8 394/63 8 216/66 9 431/266 9 253/85</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>12</td>
<td>9 294/215 10 149/53 10 184/51 11 187/51 11 258/62</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>10</td>
<td>10 145/15 10 239/54 10 153/22 10 123/14 10 118</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>8</td>
<td>8 324/33 8 288/66 8 315/50 8 296/40 8 297/40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>11</td>
<td>11 211/26 11 229/19 11 204/12 11 218/14 11 218/21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>9</td>
<td>0 0 0 0 3 3 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Table 8.9 | Number of starting points for which the SQP-Newton method converges. |
Chapter 9

Concluding Remarks

We have presented a new and robust choice for the Lagrange multipliers in the SQP method. This choice strongly utilizes a penalty parameter. This penalty parameter is computed as the solution of a constrained optimization problem whose objective function is an approximation to the Byrd and Nocedal function (4.7) and the constraint is a sufficient condition for a symmetric matrix to be positive definite. The sufficient condition is due to Tarazaga in [27] and [28]. This choice of the Lagrange multipliers preserves the local and q-quadratic convergence of the Lagrangian SQP-Newton method.

Our numerical results indicate that this new choice of the Lagrange multipliers generates a positive definite Hessian of the Lagrangian more frequently than the traditional choices (least-squares (3.4), Miele-Cragg-Levy (3.5) and the QP multiplier (3.7)). Moreover, the SQP-Newton method, with this new choice of the Lagrange multipliers, converges to local minimizers more frequently than with the traditional choices. Our numerical experiments seem to indicate that the region of local convergence is larger with our choice of the multipliers.

The parallel version of the Algorithm 5.5 seems to perform better than the sequential version of Algorithm 5.5 in most of the problems. However, this deserves further investigation.

So far, we have discussed the local properties of this choice of multipliers. In the near future we would like to embed this technique in a globalization strategy.
Bibliography


