S ystems
O ptimization
L aboratory

AD-A267 216

A Sequential Quadratic Programming Algorithm Using
An Incomplete Solution of the Subproblem
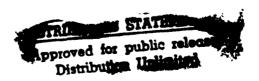
by

Walter Murray* and Francisco J. Prieto†

TECHNICAL REPORT SOL 93-3

May 1993

DTIC
ELECTE
JUL 27 1993
S E D

93-16793

Department of Operations Research
Stanford University
Stanford, CA 94305

DISTRIBUTION STATEMENT A

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-4022

A Sequential Quadratic Programming Algorithm Using
An Incomplete Solution of the Subproblem

by

Walter Murray* and Francisco J. Prieto†

TECHNICAL REPORT SOL 93-3

May 1993

A SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM USING AN INCOMPLETE SOLUTION OF THE SUBPROBLEM

WALTER MURRAY* AND FRANCISCO J. PRIETO[†]

*Systems Optimization Laboratory
Department of Operations Research
Stanford University

†Dept. de Estadistica y Econometria
Universidad Carlos III de Madrid

Abstract

We analyze sequential quadratic programming (SQP) methods to solve non-linear constrained optimization problems that are more flexible in their definition than standard SQP methods. The type of flexibility introduced is motivated by the necessity to deviate from the standard approach when solving large problems. Specifically we no longer require a minimizer of the QP subproblem to be determined or particular Lagrange multiplier estimates to be used. Our main focus is on an SQP algorithm that uses a particular augmented Lagrangian merit function. New results are derived for this algorithm under weaker conditions than previously assumed; in particular, it is not assumed that the iterates lie on a compact set.

1. Introduction

The problem of interest is the following:

$$\begin{array}{ll} \underset{x \in \Re^n}{\text{minimize}} & F(x) \\ \text{s.t.} & c(x) \geq 0, \end{array} \qquad \text{NP}$$

where $F : \Re^n \to \Re$ and $c : \Re^n \to \Re^m$. Since we shall not assume second derivatives are known, computing x^*, a point satisfying the *first-order KKT conditions* for NP is the best that can be achieved. Such points are feasible and satisfy the following conditions:

$$\nabla F(x^*) = \nabla c(x^*)^T \lambda^*, \qquad \lambda_j^* c_j(x^*) = 0 \quad j = 1, \ldots, m \qquad (1.1)$$

for some nonnegative multiplier vector $\lambda^* \in \Re^m$. Whenever the term "KKT point" is used in the following sections, what will be meant is a point satisfying the first-order KKT

1

conditions for NP. Despite this theoretical limitation we shall prefer some KKT points to others in order to try and satisfy our real purpose of finding a minimizer. For example, if the initial estimate is feasible we do not wish to converge to a nearby KKT point if at that point the objective function is higher.

We use the term *stationary point* to denote a point that is feasible and satisfies (1.1) for some multiplier vector $\lambda \in \Re^m$ that is not necessarily nonnegative.

Typically SQP algorithms generate a sequence of points $\{x_k\}$ converging to a solution, by solving at each point, x_k, a quadratic program (QP) of the form

$$\begin{array}{cl} \underset{p \in \Re^n}{\text{minimize}} & \nabla F(x_k)^T p + \frac{1}{2} p^T H_k p \\ \text{s.t.} & c(x_k) + \nabla c(x_k) p \geq 0 \end{array} \qquad \text{QP}$$

for some positive definite matrix H_k. Let p_k (referred to as the search direction) denote the unique solution to QP. We define $x_{k+1} \equiv x_k + \alpha_k p_k$, where the steplength α_k is chosen to achieve a reduction in a merit function.

SQP algorithms are viewed by many as the best approach to the solution of NP when n is small (< 200). As the size of the problem grows, usually so does the relative importance of the effort to solve QP when compared to the total effort. Indeed for many large problems the effort to solve QP dominates the total effort.

When the minimizer of QP is used to define the search direction, it is not necessary in any theoretical discussion of an SQP algorithm to define *how* the QP subproblem is solved. All implementations of SQP methods currently available use an active-set method to solve the QP subproblem. For a comprehensive survey of active-set methods see [GMW81], [Fle87] and [GMSW91]. The potential number of iterations to solve a QP using an active-set method grows exponentially with n. In practice the number of iterations grows much more slowly than exponential (if this was not the case active-set methods would be hopelessly inefficient). Nonetheless, the number of iterations required to solve a large QP is usually large. In any implementation of an SQP method it is necessary to limit the number of iterations allowed to solve a given QP subproblem. If the QP solution process is terminated prematurely the SQP algorithm may break down. It is in part for this reason that the development of SQP methods for large-scale problems has been inhibited. Even for small problems there are occasions when the number of QP iterations is excessive. Since the definition of "small" continues to increase as computers become more powerful we can expect the cost of solving the subproblems to grow in importance.

In the algorithms presented here we have endeavored to improve the efficiency of SQP methods by circumventing the need to determine the minimizer of QP. We show that a suitable search direction may be computed from information available at *any* stationary point of QP. Stationary points occur as iterates within most active-set methods to solve QP and for such methods the number of iterations to determine a stationary point increases only linearly with the size of the problem. Consequently, the search direction may be found by applying an active-set method to QP and terminating the procedure early.

It may be thought that by expending much less effort to compute the search direction, the number of iterations for the outer algorithm may increase. However, it has been observed that large numbers of QP iterations are required only when x_k is a poor approximation to x^*, that is, when the QP subproblem does not model the nonlinear problem well. We

hypothesize that a search direction based on the minimizer of such subproblems is little better than using information at a stationary point. Our preliminary results reported in Section 6 support this hypothesis.

Not solving the QP subproblem also implies that we do not know the QP multipliers, which are often used to estimate the multipliers of NP. In general, SQP methods usually use some specific estimate of the NP multipliers in the definition of the method and hence in the proof of convergence. When solving large problems specific definitions of multiplier estimates are not always computationally attractive. In our analysis we allow for flexibility in how multipliers are defined by requiring only that the multiplier estimates satisfy certain conditions.

Incomplete solutions for QP subproblems

There have been other proposals to define the search direction for an SQP algorithm other than as the minimizer of the QP subproblem. In Dembo and Tulowitzki [DT85] an algorithm is analyzed for which the search direction p_k has the property that

$$\|p_k - p_k^*\| = o(\|p_k\|),$$

where p_k^* denotes the minimizer for the k-th QP subproblem.

We follow a different approach and define a search direction for which the effort to compute it has a guaranteed bound. A different algorithm, but using the same approach, was suggested by Gurwitz and Overton [GO89]. However, no global convergence results were given for their algorithm.

In the course of solving a QP an active-set method generates iterates that are stationary points. We show that such points may be used to construct a suitable search direction. The step to the stationary point is not in general an adequate search direction. However, if the stationary point is not a minimizer then there exist nonoptimal multipliers. We show how an auxiliary direction may be constructed using information about the nonoptimal multipliers. This auxiliary direction, when combined with the step to the stationary point, gives a suitable search direction.

Terminating the QP algorithm prior to obtaining a solution impacts the SQP algorithm in a number of critical ways. Not only is the search direction different, but also the QP multipliers will not be available. The merit function of principal interest requires the definition of a search direction in the space of the multipliers. In the past, this search direction has been defined using the QP multipliers. The fact that such multipliers are positive was crucial in the analysis of these algorithms. The consequences of terminating the QP solution process early are therefore far reaching.

The remainder of this paper is organized as follows. Section 2 describes the form of the general algorithm, and the definition of the search direction. Section 3 studies the convergence properties of the algorithm; it is shown that such an algorithm is globally convergent. In Section 4 we show that the algorithm converges superlinearly. We also show that the penalty parameter used in the merit function is bounded. Section 5 considers the use of alternative merit functions. Finally, Section 6 presents numerical results obtained from an implementation that uses the merit function of principal interest.

2. Description of the algorithm

The search direction we propose could be used with most of the merit functions analyzed in the literature. However, our primary interest is the following merit function:

$$L_A(x, \lambda, s, \rho) = F(x) - \lambda^T(c(x) - s) + \tfrac{1}{2}\rho(c(x) - s)^T(c(x) - s) , \qquad (2.1)$$

where $s \geq 0$ are slack variables, and the scalar ρ is known as the penalty parameter.

This merit function was suggested by Gill et al. [GMSW86b] and is used in the SQP code NPSOL. It is similar to merit functions proposed by Wright [Wri76] and Schittkowski [Sch81]. Although our primary interest is this specific merit function, we also show (Section 5) how the ideas discussed can be extended to the use of other merit functions. The reason for our focus on this merit function is due to the success in practice of NPSOL. The merit function is also used in a new SQP code, LSSQP [Eld91], designed to solve large problems.

The search is performed on an expanded space, including the Lagrange multiplier estimates λ, and the slack variables s. The symbols p, ξ and q will be used to denote the components of the search direction on the corresponding subspaces. In this case, the value of the merit function as a function of the steplength will be denoted by

$$\phi(\alpha) \equiv L_A(x + \alpha p, \lambda + \alpha \xi, s + \alpha q, \rho). \qquad (2.2)$$

The derivative of ϕ with respect to α is denoted by ϕ'. Also, $\phi_k(\alpha)$ and $\phi'_k(\alpha)$ will be used to indicate the values of ϕ and ϕ' evaluated at $(x_k, p_k, \lambda_k, \xi_k, s_k, q_k, \rho_k)$.

The following conventions will be used in the rest of the paper:

$$g_k \equiv \nabla F(x_k), \qquad A_k \equiv \nabla c(x_k), \qquad c_k \equiv c(x_k),$$

and the symbols $\hat{A}_k$ and $\hat{c}_k$ will be used with the same meaning as A_k and c_k, but restricted to the set of active constraints at the given point. The term *active constraint* will be used to designate a constraint that is satisfied exactly at the current point ($c_j(x) = 0$ in NP, or $a_j^T p = -c_j$ in QP), and the set of all constraints active at a given point will be referred to as the *active set* at the point.

The objective function for the QP subproblem will be denoted by $\psi_k(p)$,

$$\psi_k(p) \equiv g_k^T p + \tfrac{1}{2} p^T H_k p. \qquad (2.3)$$

Sometimes, ψ will denote the function of one variable $\psi_k(\gamma) \equiv \psi_k(p + \gamma d)$.

For any vector v, the notation v^- will be used to denote the vector whose j-th element is defined as

$$v_j^- \equiv -\min(0, v_j).$$

Finally, the symbol e denotes the vector $(1, \ldots, 1)^T$, and symbols of the form β_{abc} denote fixed scalars related to properties of the problem, or the implementation of the algorithm, where "*abc*" identifies the specific scalar represented.

The algorithm

We first present an outline of the algorithm. Given H_0 positive definite, x_0 and λ_0, select $\rho_{-1} \geq 0$, $0 < \sigma < \eta < 1$, $\beta_c \geq \|c^-(x_0)\|_\infty$, $\beta_\mu \geq \|\lambda_0\|$ and $\beta_\rho > 0$.

Algorithm ETSQP

$k \leftarrow 0$

repeat

    Obtain the search direction p_k from the QP subproblem
$$\min_p \quad \psi_k(p) \equiv g_k^T p + \tfrac{1}{2} p^T H_k p$$
$$\text{s.t.} \qquad A_k p + c_k \geq 0$$

    Compute μ_k, an estimate of λ^* such that $\|\mu_k\| \leq \beta_\mu$

    $\xi_k \leftarrow \mu_k - \lambda_k$

    if $\rho_{k-1} = 0$

        Compute s_k from $(s_k)_j = \max(0, (c_k)_j)$

    else

        Compute s_k from $(s_k)_j = \max(0, (c_k)_j - (\lambda_k)_j / \rho_{k-1})$

    end if

    $q_k \leftarrow A_k p_k + c_k - s_k$

    if $\phi_k'(0) \leq -\tfrac{1}{2} p_k^T H_k p_k$

        $\rho_k \leftarrow \rho_{k-1}$

    else

$$\rho_k \leftarrow \max\Big(2\rho_{k-1}, \frac{\psi_k(p_k) + (2\lambda_k - \mu_k)^T (c_k - s_k)}{\|c_k - s_k\|^2}, \beta_\rho\Big)$$

    end if

    if $\phi_k(1) \leq \phi_k(0) + \sigma \phi_k'(0)$

        $\hat{\alpha} \leftarrow 1$

    else

        Select $\hat{\alpha} \in (0, 1)$ to satisfy
$$\phi_k(\hat{\alpha}) \leq \phi_k(0) + \sigma \hat{\alpha} \phi_k'(0), \quad |\phi_k'(\hat{\alpha})| \leq -\eta \phi_k'(0)$$

    end if

    while $c(x_k + \hat{\alpha} p_k) \not\geq -\beta_c e$ **or** $\phi_k(\hat{\alpha}) > \phi_k(0) + \sigma \hat{\alpha} \phi_k'(0)$ **do**

        $\hat{\alpha} \leftarrow \hat{\alpha}/2$

    end do

    $\alpha_k \leftarrow \hat{\alpha}$

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} \leftarrow \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} + \alpha_k \begin{pmatrix} p_k \\ \xi_k \end{pmatrix}$$

    Compute g_{k+1}, A_{k+1} and c_{k+1}

    Update H_k to form H_{k+1}

    $k \leftarrow k + 1$

until *convergence*

The following are some comments on the steps of the algorithm.

- At each point x_k, we form the QP subproblem

$$\underset{p \in \Re^n}{\text{minimize}} \quad g_k^T p + \tfrac{1}{2} p^T H_k p \tag{2.4a}$$

$$\text{subject to} \quad A_k p \geq -c_k, \tag{2.4b}$$

and determine a stationary point for QP, that is, a point $\tilde{p}_k$ satisfying

$$g_k + H_k \tilde{p}_k = A_k^T \pi_k, \tag{2.5a}$$

$$A_k \tilde{p}_k + c_k \geq 0, \qquad \pi_k^T (A_k \tilde{p}_k + c_k) = 0, \tag{2.5b}$$

for some vector $\pi_k \in \Re^m$.

From information available at the stationary point we construct a search direction p_k and μ_k an estimate of λ^*. The precise conditions that p_k and μ_k need to satisfy are given later in this section. If $p_k = 0$, we set $\lambda_k = \mu_k$ and terminate. Otherwise, we compute the search direction in the space of the multiplier estimates ξ_k as

$$\xi_k = \mu_k - \lambda_k. \tag{2.6}$$

- The slack variables s_k are computed from

$$(s_k)_j = \begin{cases} \max\left(0, (c_k)_j\right) & \text{if } \rho_{k-1} = 0, \\ \max\left(0, (c_k)_j - \dfrac{(\lambda_k)_j}{\rho_{k-1}}\right) & \text{otherwise.} \end{cases} \tag{2.7}$$

These values minimize the merit function (2.1) at $(x_k, \lambda_k, \rho_{k-1})$ with respect to the slack variables.

The slack variables s_k appear in the merit function (2.1) as part of the term $c_k - s_k$. From (2.7), this term takes the value

$$(c_k)_j - (s_k)_j = \begin{cases} \min\left(0, (c_k)_j\right) & \text{if } \rho_{k-1} = 0, \\ \min\left((c_k)_j, \dfrac{(\lambda_k)_j}{\rho_{k-1}}\right) & \text{otherwise.} \end{cases} \tag{2.8}$$

We shall require the following inequality:

$$\|c_k^-\| \leq \|c_k - s_k\|. \tag{2.9}$$

To simplify the notation in the justification of this result, we drop the subscript k.

If $c_j - s_j = c_j$ then clearly $|c_j - s_j| = |c_j| \geq |c_j^-|$.

If $c_j - s_j \neq c_j$ and $c_j \geq 0$, then $c_j^- = 0 \leq |c_j - s_j|$. Otherwise, $c_j - s_j \neq c_j$ and $c_j < 0$. From (2.8) we get $c_j - s_j < c_j < 0$, and hence $|c_j - s_j| > |c_j| \geq |c_j^-|$. We have shown $|c_j^-| \leq |c_j - s_j|$ under all circumstances, implying (2.9).

- The search direction in the space of the slack variables q_k is set to the vector of slack variables for the QP subproblem, i.e.

$$q_k = A_k p_k + c_k - s_k. \tag{2.10}$$

For a linear constraint this choice keeps the corresponding slack variable at its optimum value.

- The penalty parameter will not be modified if the condition

$$\phi'_k(0) \leq -\tfrac{1}{2}p_k^T H_k p_k, \tag{2.11}$$

is satisfied, where $\phi_k(\alpha)$ is defined in (2.2). Otherwise, we define the penalty parameter as

$$\rho_k = \max(2\rho_{k-1}, \hat{\rho}_k, \beta_\rho), \tag{2.12}$$

where β_ρ is some positive constant,

$$\hat{\rho}_k \equiv \frac{\psi_k(p_k) + (2\lambda_k - \mu_k)^T(c_k - s_k)}{\|c_k - s_k\|^2}, \tag{2.13}$$

and ψ_k was defined in (2.3). It will be shown that the definition (2.12) ensures that (p_k, ξ_k, q_k) is a sufficient descent direction for the merit function, in the sense that condition (2.11) holds for this value of the penalty parameter.

- The steplength $\alpha_k > 0$ is computed to reduce $\phi_k(\alpha)$ while keeping the constraint violation bounded. The termination conditions for the linesearch are as follows:

If

$$\phi_k(1) - \phi_k(0) \leq \sigma \phi'_k(0), \tag{2.14}$$

set $\hat{\alpha} = 1$. Otherwise, find an $\hat{\alpha} \in (0, 1)$ such that

$$\phi_k(\hat{\alpha}) - \phi_k(0) \leq \sigma \hat{\alpha} \phi'_k(0) \tag{2.15a}$$

$$\phi'_k(\hat{\alpha}) \geq \eta \phi'_k(0), \tag{2.15b}$$

where $0 < \sigma < \eta < 1$.

If the condition

$$c(x_k + \hat{\alpha} p_k) \geq -\beta_c e \tag{2.16}$$

holds, we define $\alpha_k = \hat{\alpha}$; otherwise we compute α_k by performing a backtracking linesearch from $\hat{\alpha}$ until (2.15a) and (2.16) are both satisfied. It will be shown later that such a steplength always exists, and that Algorithm **ETSQP** is well defined. This definition of the steplength ensures that $c(x_k) \geq -\beta_c e$ for all k. A more sophisticated algorithm could be used to determine α_k when (2.16) does not hold. However, we anticipate such events will be rare.

- Finally, x_k and λ_k are updated from

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} + \alpha_k \begin{pmatrix} p_k \\ \xi_k \end{pmatrix}. \tag{2.17}$$

The definition of the search direction

At each iteration of **ETSQP** an inner iteration is performed to determine the search direction by solving the QP subproblem (2.4) using an active-set method. The following is an outline of a suitable algorithm to determine the search direction. The outer iteration subscript has been omitted, and the subscript i refers to the inner iterations.

We assume that positive constants β_p, β_b, γ_M have been defined.

Algorithm SD

Compute p_0 satisfying:

$$Ap_0 + c \geq 0, \qquad \|p_0\| \leq \beta_p\|c^-\|, \qquad g^T p_0 \leq \beta_p\|c^-\|$$

Form $\hat{A}_0$, the active-set matrix at p_0, as the set of all rows in A corresponding to active QP constraints at p_0

$i \leftarrow 0$

repeat

    Compute $\bar{p}_i$ from $\begin{pmatrix} H & \hat{A}_i^T \\ \hat{A}_i & 0 \end{pmatrix} \begin{pmatrix} \bar{p}_i \\ -\pi_i \end{pmatrix} = \begin{pmatrix} -g - H p_i \\ 0 \end{pmatrix}$

    $\gamma_i \leftarrow \min\left(1, \inf_j\left\{ -\dfrac{c_j + a_j^T p_i}{a_j^T \bar{p}_i} \mid a_j^T \bar{p}_i < 0 \right\}\right)$

    $p_{i+1} \leftarrow p_i + \gamma_i \bar{p}_i$

    Set $\hat{A}_{i+1}$ to be the active-set matrix at p_{i+1}

    $i \leftarrow i + 1$

until (p_i, π_i) satisfy (2.5)

$\tilde{p} \leftarrow p_i$

$\pi \leftarrow \pi_i$

if $\pi \geq 0$

    $p \leftarrow \tilde{p}$

else

    Define v to satisfy: $\|v\| = 1$, $v \geq 0$, $v_j \pi_j \leq 0$ $\forall j$, $v^T \pi \leq \beta_b \min_j \pi_j$

    Compute $\bar{d}$ by solving: $\min\{\bar{d}^T \bar{d} \mid \hat{A}_i \bar{d} = v\}$

    $d \leftarrow \bar{d}/\|\bar{d}\|$

    $\gamma \leftarrow \min\left(-\dfrac{(g + H\tilde{p})^T d}{d^T H d}, \inf_j\left\{ -\dfrac{c_j + a_j^T \tilde{p}}{a_j^T d} \mid a_j^T d < 0 \right\}, \gamma_M\right)$

    if $\|\tilde{p} + \gamma d\| > \|\tilde{p}\|$

        $p \leftarrow \tilde{p} + \gamma d$

    else

        $p \leftarrow \tilde{p}$

    end if

end if

Some comments on this procedure are presented below.

- An initial feasible point p_0 of the QP subproblem is obtained.

  When the minimizer of the QP is used as the search direction, then, given the uniqueness of p, the choice of p_0 is irrelevant. If we determine the search direction from a stationary point that is not a minimizer, the sequence of stationary points that we compute depends directly on the value of p_0. We wish to define the initial point in such a manner that *all* stationary points are satisfactory points at which to terminate the solution process. It will be seen that the following conditions on p_0 are sufficient to ensure our objective.

  - For some constant $\beta_p > 0$,

    $$\|p\| \le \beta_p \|c^-\| \quad \text{and} \quad g^T p_0 \le \beta_p \|c^-\|. \tag{2.18}$$

- A sequence of feasible descent steps are taken, for example, by first computing the step $\bar{p}_i$ to the minimizer of the QP on the current working set as the solution of the system of equations

  $$\begin{pmatrix} H & \hat{A}_i^T \\ \hat{A}_i & 0 \end{pmatrix} \begin{pmatrix} \bar{p}_i \\ -\pi_i \end{pmatrix} = \begin{pmatrix} -g - H p_i \\ 0 \end{pmatrix}, \tag{2.19}$$

  where p_i is the current estimate. A step γ_i is taken, where γ_i is obtained as either one or the step to the nearest constraint,

  $$\gamma_i = \min\left(1, \inf_j\left\{-\frac{c_j + a_j^T p_i}{a_j^T \bar{p}_i} \mid a_j^T \bar{p}_i < 0\right\}\right). \tag{2.20}$$

  The QP algorithm may be terminated at *any* stationary point $\bar{p}$. (Algorithm **SD** is terminated at the first stationary point.) It will be seen in the proofs that to always use $\bar{p}$ as the search direction will not in general ensure convergence.

- If $\bar{p}$ is the minimizer of the QP subproblem, that is, if $\pi \ge 0$, the search direction p is defined as $p \equiv \bar{p}$, otherwise

  $$p \equiv \begin{cases} \bar{p} + \bar{\gamma}d & \text{if } \|\bar{p}\| < \|\bar{p} + \bar{\gamma}d\|, \\ \bar{p} & \text{otherwise}, \end{cases} \tag{2.21}$$

  where the vector d and the scalar $\bar{\gamma}$ are computed with the following properties:

  - d is feasible with respect to the active QP constraints at $\bar{p}$, $\hat{A}_i d \ge 0$, and it has unit norm, $\|d\| = 1$.

  - The rate of descent along d is sufficiently large. Specifically, we require

    $$(H\bar{p} + g)^T d \le \beta_b \min_j \pi_j, \tag{2.22}$$

  for some positive constant β_b.

There are many procedures for computing a sui'able vector d; we now describe one such procedure (see algorithm **SD**). It proceeds by defining an auxiliary vector v with the following properties

$$\|v\| = 1, \quad v \geq 0, \quad v_j \pi_j \leq 0 \; \forall j, \quad v^T \pi \leq \beta_b \min_j \pi_j;$$

such a vector can be obtained for example by letting

$$v_j = \begin{cases} 1 & \text{if } \pi_j < 0 \\ 0 & \text{otherwise} \end{cases}.$$

We then compute $\bar{d}$ the least-length solution of $\hat{A}_i y = v$ and set

$$d = \bar{d}/\|\bar{d}\|. \tag{2.23}$$

— The scalar $\tilde{\gamma}$ is given by

$$\tilde{\gamma} = \min(\bar{\gamma}, \hat{\gamma}, \gamma_M), \tag{2.24}$$

where γ_M is a specified upper bound on the steplength,

$$\bar{\gamma} = \inf_j \left\{ -\frac{c_j + a_j^T \tilde{p}}{a_j^T d} \mid a_j^T d < 0 \right\}, \tag{2.25}$$

is the largest feasible step from $\tilde{p}$ along d, and

$$\hat{\gamma} = -\frac{(g + H\tilde{p})^T d}{d^T H d}, \tag{2.26}$$

is the step to the minimizer of $\psi(\tilde{p} + \gamma d)$.

The multiplier estimates

Equation (2.6) defining the search direction on the multiplier space ξ_k requires the computation of an estimate μ_k for the Lagrange multipliers. The estimates $\{\mu_k\}$ are then used to update $\{\lambda_k\}$, the Lagrange multiplier estimate used in the merit function. To allow flexibility in algorithm design we have chosen to specify conditions on the multipliers estimates μ_k rather than give explicit definitions.

It will be shown that the following conditions on μ_k are sufficient to ensure that the algorithm is globally convergent.

MC1. The estimates μ_k are uniformly bounded in norm, that is $\|\mu_k\| \leq \beta_\mu < \infty$.

MC2. The complementarity condition $\mu_k^T(A_k p_k + c_k) = 0$ is satisfied at all iterations.

We may satisfy these conditions by choosing $\mu_k = 0$. Condition **MC2** is made for convenience; condition **MC1** and the form in which the multiplier estimates are updated imply that $\{\lambda_k\}$ are uniformly bounded.

Lemma 2.1. *If condition* **MC1** *holds, then* $\|\lambda_k\| \leq \beta_\mu$ *for all* k.

Proof. The proof is by induction. We select β_μ to satisfy $\|\lambda_0\| \leq \beta_\mu$. From (2.17),

$$\lambda_{k+1} = \lambda_k + \alpha_k(\mu_k - \lambda_k), \quad k \geq 0. \tag{2.27}$$

Using norm inequalities and $0 < \alpha_k \leq 1$, we have

$$\|\lambda_{k+1}\| \leq \alpha_k\|\mu_k\| + (1 - \alpha_k)\|\lambda_k\| \leq \alpha_k\beta_\mu + (1 - \alpha_k)\beta_\mu = \beta_\mu,$$

as required. ∎

Second-order information

We choose the matrices $\{H_k\}$ to be positive definite and bounded, with bounded condition number. In practice, such matrices may be generated (see [GMSW86a]) by updating a quasi-Newton approximation to the Hessian of the Lagrangian function or the Hessian of the augmented Lagrangian function in each iteration together with certain safeguards (for example, if the factors of H_k are updated, by enforcing bounds on the size of the elements, and ensuring sufficiently positive diagonal elements). These conditions can be written as follows:

HC1. $\beta_{lvH} < \infty$ is the largest eigenvalue of $\{H_k\}$.

HC2. $\beta_{svH} > 0$ is the smallest eigenvalue of $\{H_k\}$.

3. Global convergence results

The results in this section establish global convergence properties for algorithm **ETSQP**. We first introduce the assumptions under which we shall show convergence, and then we prove the following results:

- The iterates $\{x_k\}$ lie on a compact set.

    - In Lemma 3.1 we show that the quantities associated with the algorithm are well defined at all points.

    - In Lemma 3.2 it is shown that if $\|x_k\|$ is large then $\|p_k\|$ cannot be arbitrarily small.

    - In Lemma 3.3 we show that p computed using algorithm **SD** satisfies

$$\psi(p) \equiv g^T p + \tfrac{1}{2}p^T H p \leq -\beta_1 p^T H p + \beta_2\|c - s\|,$$

    where β_1 and β_2 are positive constants.

    - Lemma 3.4 proves that the sequence $\{x_k\}$ lies on a compact set.

    - Lemma 3.5 shows that the sequence $\{p_k\}$ also remains bounded.

- The sequence $\{\|p_k\|\}$ dominates the sequence $\{\|x_k - x^*\|\}$, where x^* denotes a KKT point closest to x_k. The main implication of this result is that $\|p_k\| \to 0$ is sufficient to ensure that $x_k \to x^*$, a KKT point of NP.

- It is shown in Lemma 3.6 that, under the assumptions we make, the KKT points for problem NP are isolated.

- Lemma 3.7 introduces another preliminary result, proving that if $p_k \to 0$ along a subsequence then along this subsequence $\|x_k - x^*\| \to 0$, where x^* is a KKT point for NP nearest to x_k. Moreover, for large enough k, p_k is the minimizer of the QP subproblem, and the correct active set at x^* is identified.

- The proof that $\|p_k\|$ dominates $\|x_k - x^*\|$ is given in Lemma 3.8.

- **Bounds on the growth of the penalty parameter ρ_k.**

  We cannot prove that ρ_k will remain bounded in the algorithm without stronger conditions on the multiplier estimate μ_k, but we can show that its growth is bounded by certain quantities related with the algorithm, and that is enough to prove convergence.

  - We show in Lemma 3.9 that at all the iterations where the penalty parameter is modified the following bounds hold,

  $$\rho_k \|c_k - s_k\| \leq N \quad \text{and} \quad \rho_k \|p_k\|^2 \leq N.$$

  - In Lemma 3.10 and Lemma 3.11 we show that similar inequalities hold at all iterations.

- **The steplength α_k is bounded away from zero if we are not close to a solution.**

  - We first need a bound on the second derivatives of $\phi(\alpha)$. In Lemma 3.12 we prove that $\phi_k(\alpha_k) \leq N$ for some positive constant N.

  - In Lemma 3.13 we show that, if $\|p_k\|$ is large enough, there exists a value $\bar{\alpha} > 0$ independent of the iteration such that $\alpha_k \geq \bar{\alpha}$.

- **In Theorem 3.1 we show that $x_k \to x^*$.**

- **Finally, we prove that $\lambda_k \to \lambda^*$.**

  - This result requires stronger conditions on the multiplier estimate μ_k than just **MC1** and **MC2**. We start by introducing a third condition **MC3**.

  - Lemma 3.14 strengthens the result in Lemma 3.13 showing that, under the new conditions on the multipliers, α_k is uniformly bounded away from zero.

  - In Theorem 3.2 we show that $\lambda_k \to \lambda^*$.

Assumptions

Some of the following assumptions make use of the concepts of stationary points and KKT points at infinity. We will say that NP has a stationary point at infinity if there exist sequences $\{x_k\}$ and $\{\eta_k\}$ such that $\|x_k\| \to \infty$ and/or $\|\eta_k\| \to \infty$, and

$$c_k^- \to 0, \quad A_k^T \eta_k - g_k \to 0, \quad \eta_k^T c_k \to 0.$$

If in addition to these conditions we also have $\eta_k^- \to 0$, then we have a KKT point at infinity.

We make the following assumptions:

A1. For some constant $\beta_c > 0$, the global minimum of the problem

$$\underset{x \in \Re^n}{\text{minimize}} \quad F(x)$$
$$\text{s.t.} \quad c(x) \geq -\beta_c e,$$

is bounded below.

A2. There exist no KKT points at infinity for problem NP.

A3. F, c_j and their first and second derivatives are continuous and uniformly bounded in norm on a compact set.

A4. The Jacobian corresponding to the active constraints at all KKT points has full rank.

A5. A feasible point p_{k_0} exists to all the QP subproblems, satisfying

$$\|p_{k_0}\| \leq \beta_p \|c_k^-\| \quad \text{and} \quad g_k^T p_{k_0} \leq \beta_p \|c_k^-\|$$

for some constant $\beta_p > 0$.

A6. Strict complementarity holds at all stationary points of NP, including stationary points at infinity, if they exist.

A7. The reduced Hessian of the Lagrangian function is nonsingular at all KKT points.

The larger the value of β_c, the stronger is assumption **A1**. There will be problems, for example $F(x) = f(x)^T f(x)$, where it is known a priori that assumption **A1** holds with $\beta_c = \infty$. Also, if **A1** does not hold with $\beta_c = 0$ then it is possible for any reasonable algorithm to diverge.

Assumption **A5** imposes conditions on the initial point for the QP. It is possible that no point satisfies these conditions; this would be the case for example if one of the QP subproblems generated by the algorithm is not feasible. Nevertheless, by introducing an additional variable it is possible to construct a modified problem for which this condition is satisfied trivially. Consider the problem

$$\underset{(x,\bar{x}) \in \Re^{n+1}}{\text{minimize}} \quad \mathcal{F}(x,\bar{x}) \equiv (1 - \omega)F(x) + \omega\bar{x}$$
$$\text{s.t.} \quad c(x) + \bar{x}e \geq 0 \quad \text{and} \quad \bar{x} \geq 0,$$

where $\bar{x} \in \Re$ and $\omega \in [0,1]$. The KKT points for this problem are also KKT points for NP if NP is feasible and ω is sufficiently close to one. The modified problem is always feasible, and the corresponding QP subproblem takes the form

$$\underset{(p,\bar{p}) \in \Re^{n+1}}{\text{minimize}} \quad (1 - \omega)g_k^T p + \omega\bar{p} + \tfrac{1}{2} \begin{pmatrix} p^T & \bar{p} \end{pmatrix} H_k \begin{pmatrix} p \\ \bar{p} \end{pmatrix}$$
$$\text{s.t.} \quad c_k + A_k p + \bar{x}_k e + \bar{p}e \geq 0$$
$$\bar{x}_k + \bar{p} \geq 0.$$

For this QP subproblem the point

$$p_0 = \begin{pmatrix} p \\ \bar{p} \end{pmatrix} = \begin{pmatrix} 0 \\ \|(c_k + \bar{x}_k e)^-\|_\infty \end{pmatrix}$$

is feasible since we can ensure that $\tilde{x}_k \geq 0$. Therefore there always exists a feasible point that satisfies **A5** with $\beta_p = 1$ since $\|p_0\| = \|(c_k + \tilde{x}_k e)^-\|_\infty$ and

$$\nabla \mathcal{F}_k^T p_0 = \left(\begin{array}{cc} (1-\omega)g_k^T & \omega \end{array} \right) \left(\begin{array}{c} p \\ \bar{p} \end{array} \right) = \omega \|(c_k + \tilde{x}_k e)^-\|_\infty \leq \|(c_k + \bar{x}_k e)^-\|_\infty,$$

implying that Assumption **A5** is redundant for the modified problem.

Existence of the iterates

We start by showing that all the quantities associated with the algorithm are well defined. In particular, we show that the choice of penalty parameter ensures (2.11) is satisfied and that the steplength exists.

Lemma 3.1. *Under assumptions* **A3**, **A5** *and conditions* **HC1**, **HC2**, *the procedures given in the algorithm to compute the values of the penalty parameter ρ_k and the steplength α_k are well defined.*

Proof. We drop the subscript k denoting the iteration number, to simplify the notation.

Consider the gradient of the merit function L_A, defined in (2.1), with respect to x, λ and s,

$$\nabla L_A(x, \lambda, s) = \left(\begin{array}{c} g(x) - A(x)^T \lambda + \rho A(x)^T (c(x) - s) \\ -(c(x) - s) \\ \lambda - \rho(c(x) - s) \end{array} \right). \tag{3.1}$$

It follows from (2.6), (2.10) and (2.2) that $\phi'(0)$ is given by

$$\begin{aligned}
\phi'(0) &= p^T g - p^T A^T \lambda + \rho\, p^T A^T (c - s) - (c - s)^T \xi + \lambda^T q - \rho q^T (c - s) \\
&= p^T g + (2\lambda - \mu)^T (c - s) - \rho \|c - s\|^2,
\end{aligned} \tag{3.2}$$

where g, A, and c are evaluated at x.

If $\|c - s\| = 0$, from (2.9) and (2.18) we have $p_0 = 0$, and since $\psi(p) = p^T g + \frac{1}{2} p^T H p \leq \psi(p_0) = 0$ it follows that

$$\phi'(0) = p^T g \leq -\tfrac{1}{2} p^T H p,$$

implying that ρ does not need to be modified.

If $\|c - s\| > 0$, we obtain from (3.2) that for $\rho = \hat{\rho}$ (defined in (2.13))

$$\phi'(0) = g^T p + (2\lambda - \mu)^T (c - s) - \hat{\rho} \|c - s\|^2 = -\tfrac{1}{2} p^T H p,$$

which implies the desired descent condition (2.11) is satisfied for all $\rho \geq \hat{\rho}$.

An immediate consequence of (2.11) and the properties of H_k is the following bound on the directional derivative:

$$\phi'_k(0) \leq -\tfrac{1}{2} \beta_{svH} \|p_k\|^2. \tag{3.3}$$

It follows from the procedure to increase the value of the penalty parameter (see (2.12)) that $\rho_k \to \infty$ if and only if the parameter is increased an infinite number of times.

We also need to prove that the value of α_k introduced in the algorithm is well defined. We show that if condition (2.14) is not satisfied, a steplength $\hat{\alpha}_k \in (0,1)$ that satisfies conditions (2.15) always exists (see, for example, Moré and Sorensen [MS84]).

Define the functions

$$\chi(\alpha) \equiv \phi(\alpha) - \phi(0) - \sigma\alpha\phi'(0)$$
$$\zeta(\alpha) \equiv \phi'(\alpha) - \eta\phi'(0),$$

and note that from $\sigma < \eta$ and $\phi'(0) < 0$, implied by (2.11), we have

$$\chi'(\alpha) = \phi'(\alpha) - \sigma\phi'(0) < \phi'(\alpha) - \eta\phi'(0) = \zeta(\alpha), \tag{3.4}$$

for any α.

If (2.14) does not hold,

$$\phi(1) - \phi(0) > \sigma\phi'(0) \quad \Rightarrow \quad \chi(1) > 0,$$

and we also have $\chi(0) = 0$. From these two results and the mean-value theorem, there will be a point $\breve{\alpha} \in [0,1]$ such that $\chi'(\breve{\alpha}) > 0$, and from (3.4), $\zeta(\breve{\alpha}) > 0$.

From $\phi'(0) < 0$ we have $\zeta(0) < 0$, and the continuity of ζ (assumption **A3**) will imply the existence of a zero of ζ in $(0, \breve{\alpha})$. Let $\hat{\alpha}$ denote the smallest point in $(0, \breve{\alpha})$ such that $\zeta(\hat{\alpha}) = 0$, that is,

$$\phi'(\hat{\alpha}) = \eta\phi'(0), \tag{3.5}$$

and condition (2.15b) is satisfied at $\hat{\alpha}$.

From $\zeta(0) < 0$ we must have

$$\zeta(\alpha) < 0 \quad \forall\alpha \in [0, \hat{\alpha}) \quad \Leftrightarrow \quad \phi'(\alpha) < \eta\phi'(0) \quad \forall\alpha \in [0, \hat{\alpha}), \tag{3.6}$$

implying that condition (2.15b) is not satisfied for any point in $[0, \hat{\alpha})$.

Finally, from (3.4) and (3.6), we have

$$\chi'(\alpha) < 0 \quad \forall\alpha \in [0, \hat{\alpha}),$$

and this together with $\chi(0) = 0$ implies $\chi(\hat{\alpha}) < 0$, that is,

$$\phi(\hat{\alpha}) - \phi(0) < \sigma\hat{\alpha}\phi'(0), \tag{3.7}$$

showing that $\hat{\alpha}$ satisfies both conditions (2.15) simultaneously.

We still need to consider condition (2.16). For the function $h(\alpha) \equiv c(x + \alpha p) + \beta_c e$ we have from (2.4b)

$$h'(0) = Ap \geq -c.$$

If $-\frac{1}{2}\beta_c \geq c_j \geq -\beta_c$, we have $h_j(0) \geq 0$ and $h'_j(0) \geq \frac{1}{2}\beta_c > 0$; if $c_j \geq -\frac{1}{2}\beta_c$ then $h_j(0) \geq \frac{1}{2}\beta_c > 0$ and in any case there exists a value $\bar{\alpha} > 0$ such that $h_j(\alpha) \geq 0$ (implying $c_j(x+\alpha p) \geq -\beta_c$) for all j and all $\alpha \in [0, \bar{\alpha}]$, implying that for $\alpha \in [0, \min(\hat{\alpha}, \bar{\alpha})]$ both conditions (2.15a) and (2.16) hold simultaneously. ∎

This lemma implies that all the quantities associated with the algorithm are well defined.

Boundedness of the iterates

To prove global convergence we show first that if assumptions **A1** and **A2** hold, all points in the sequence $\{x_k\}$ generated by the algorithm lie on a compact set. We start by showing that for $\|x_k\|$ large enough we cannot have $\|p_k\|$ arbitrary small.

Lemma 3.2. *Under assumptions **A2** and **A6** and condition **HC1**, there exist positive constants M and ϵ such that $\|x_k\| \geq M \quad \Rightarrow \quad \|p_k\| \geq \epsilon$.*

Proof. Assume this result does not hold. Then, for any positive constants M and ϵ we can find iterates such that $\|x_k\| \geq M$ and $\|p_k\| < \epsilon$, and we could construct a sequence $\{x_k\}$, and its associated sequence $\{p_k\}$, along which $\|x_k\| \to \infty$ and $\|p_k\| \to 0$. For this sequence, from $\|p_k\| \to 0$ and (2.4b), we must have $\|c_k^-\| \to 0$. Also, from the definition of p_k, (2.21), it must hold that $\|\tilde{p}_k\| \to 0$, and from (2.5a) and **MC1**, we must have

$$\|A_k^T \pi_k - g_k\| = \|H_k \tilde{p}_k\| \to 0.$$

Since $\|p_k\| \to 0$ and $\|\tilde{p}_k\| \to 0$, using (2.21) and $\|d_k\| = 1$, we also have either $\bar{\gamma}_k \to 0$ or $\bar{\gamma}_k = 0$ for k large enough. It then follows from (2.24) that either $\min(\bar{\gamma}_k, \tilde{\gamma}_k) \to 0$ or $\bar{\gamma}_k = \tilde{\gamma}_k = 0$ for k large enough. If $\bar{\gamma}_k \to 0$ along a subsequence, then (2.25) implies for some constraint j that $(\pi_k)_j \to 0$ and $c_j(x_k) \to 0$, but this would contradict assumption **A6**. If $\hat{\gamma}_k \to 0$ along a subsequence, then from (2.26) and (2.22) we get $\|\pi_k^-\| \to 0$.

The properties of this sequence,

$$\|x_k\| \to \infty, \quad \|c_k^-\| \to 0, \quad \|A_k^T \pi_k - g_k\| \to 0,$$

and either $\|\pi_k^-\| \to 0$ or $\|\pi_k^-\| = 0$ for k large enough imply there exists a KKT point at infinity, which violates assumption **A2**, so the lemma must hold. ∎

Another result we need for the compactness proof is a bound on the value of the QP objective function at the incomplete solution for the QP.

Lemma 3.3. *Under assumption **A5** and conditions **HC1**, **HC2**, for p computed by algorithm **SD** there exist constants $\beta_1 > 0$ and $\beta_2 > 0$ such that*

$$\psi(p) \equiv g^T p + \tfrac{1}{2} p^T H p \leq -\beta_1 p^T H p + \beta_2 \|c - s\|.$$

Proof. The result will be shown by considering first the initial point for the QP, p_0, and then the descent achieved in each QP iteration.

By definition

$$\psi(p_0) = -\tfrac{1}{2} p_0^T H p_0 + g^T p_0 + p_0^T H p_0.$$

Since $\|p_0\| \leq \beta_p \|c^-\|$ and $g^T p_0 \leq \beta_p \|c^-\|$ (assumption **A5**), condition **HC1** on H implies

$$\psi(p_0) \leq -\tfrac{1}{2} p_0^T H p_0 + \beta_p \|c^-\| + \beta_{lvH} \beta_p^2 \|c^-\|^2. \tag{3.8}$$

Consider the quadratic function $b\gamma + \tfrac{1}{2} c\gamma^2$, where $b < 0$ and $c > 0$; then for all $\gamma \in [0, -b/c]$ (between 0 and the minimizer), we have

$$\gamma \leq -\frac{b}{c} \quad \Rightarrow \quad \gamma(b + c\gamma) \leq 0 \quad \Rightarrow \quad b\gamma + \tfrac{1}{2} c\gamma^2 \leq -\tfrac{1}{2} c\gamma^2. \tag{3.9}$$

The change in the QP objective function at any intermediate QP iteration i can be represented as

$$\psi(p_{i+1}) - \psi(p_i) = \tfrac{1}{2}\gamma_i^2 d_i^T H d_i + \gamma_i(g + H p_i)^T d_i, \tag{3.10}$$

where d_i is used to denote the QP step obtained from (2.19) or the final step d defined in (2.23), and γ_i is a feasible steplength bounded by the steplength to the minimizer along v_i, as defined in (2.20) or (2.24). We have $d_i^T H d_i > 0$ (from condition **HC2**) and $(g + H p_i)^T d_i < 0$ (from condition (2.22) and $\min_j \pi_j < 0$), implying that we can apply the bound (3.9) to (3.10) to obtain

$$\psi(p_{i+1}) - \psi(p_i) \le -\tfrac{1}{2}\gamma_i^2 d_i^T H d_i. \tag{3.11}$$

If we have taken N iterations to compute p (the search direction), by adding the inequalities (3.11) for $i = 0, \dots, N$ and using (3.8) we obtain

$$\psi(p) = \psi(p_0) + \sum_{i=1}^{N}(\psi(p_i) - \psi(p_{i-1}))$$

$$\le -\tfrac{1}{2}(p_0^T H p_0 + \sum_{i=1}^{N}\gamma_i^2 d_i^T H d_i) + \beta_p \|c^-\| + \beta_{lvH}\beta_p^2\|c^-\|^2. \tag{3.12}$$

We can use the convexity of the function $p^T H p$ (implied by property **HC2**) to write

$$p_0^T H p_0 + \sum_{i=1}^{N}\gamma_i^2 d_i^T H d_i \ge \frac{1}{N+1}\left(p_0 + \sum_{i=1}^{N}\gamma_i d_i\right)^T H\left(p_0 + \sum_{i=1}^{N}\gamma_i d_i\right) = \frac{1}{N+1}p^T H p.$$

This result together with (3.12) implies

$$\psi(p) \le -\frac{1}{2(N+1)}p^T H p + \beta_p\|c^-\| + \beta_{lvH}\beta_p^2\|c^-\|^2. \tag{3.13}$$

Since $c^- \ge \beta_c e$ the desired result follows from this inequality and (2.9). ∎

We can now prove the main result of this section.

Lemma 3.4. *Under assumptions* **A1**, **A2**, **A3**, **A5** *and* **A6**, *and conditions* **MC1**, **HC1** *and* **HC2**, *the sequence* $\{x_k\}$ *generated by the algorithm lies on a compact set.*

Proof. First we show the set of points at which the penalty parameter is modified lies on a compact set. If ρ_k remains bounded it follows from the manner the penalty parameter is modified, (2.12), that there is only a finite set of such points. Therefore we need only study the case when $\rho_k \to \infty$. Consider the iterations k where the penalty parameter is modified. From condition **MC1** and the boundedness of the multiplier estimates λ_k (Lemma 2.1), we have

$$\|2\lambda_k - \mu_k\| \le 2\|\lambda_k\| + \|\mu_k\| \le 3\beta_\mu. \tag{3.14}$$

This result, together with the definition of the penalty parameter (2.13), and Lemma 3.3 gives

$$\rho_k\|c_k - s_k\|^2 \le g_k^T p_k + \tfrac{1}{2}p_k^T H_k p_k + (2\lambda_k - \mu_k)^T(c_k - s_k)$$

$$\le (\beta_1 + 3\beta_\mu)\|c_k - s_k\| - \beta_1 p_k^T H_k p_k. \tag{3.15}$$

As we have assumed $\rho_k \to \infty$, (3.15) implies $\|c_k - s_k\| \to 0$, and from (2.9) also $\|c_k^-\| \to 0$. From Lemma 3.3 and (3.14) we have

$$\omega_k \equiv g_k^T p_k + (2\lambda_k - \mu_k)^T (c_k - s_k) \tag{3.16a}$$

$$\leq -\tfrac{1}{2} p_k^T H_k p_k - \beta_1 p_k^T H_k p_k + (\beta_1 + 3\beta_\mu)\|c_k - s_k\|. \tag{3.16b}$$

If $\|p_k\| \geq \epsilon > 0$ along an infinite subsequence, then it follows from $\|c_k - s_k\| \to 0$ and MC2 that there exists an index K such that for all $k \geq K$ in the subsequence,

$$(\beta_p + 3\beta_\mu)\|c_k - s_k\| \leq \beta_1 p_k^T H_k p_k.$$

From (3.16b) we obtain the following bound on ω_k,

$$\omega_k \leq -\tfrac{1}{2} p_k^T H_k p_k, \tag{3.17}$$

for $k \geq K$. From (3.16a) and the bounds (3.17) and (3.2), we have for sufficiently large k

$$\phi_k'(0) = \omega_k - \rho_k \|c_k - s_k\|^2 \leq \omega_k \leq -\tfrac{1}{2} p_k^T H_k p_k.$$

This last inequality implies that ρ_k is not modified for all $k \geq K$, which contradicts our assumption that the penalty parameter was modified an infinite number of times.

We have shown that $\|p_k\| \to 0$ along the subsequence at which the penalty parameter is modified. The boundedness of $\|x_k\|$ along this subsequence follows from Lemma 3.2.

We now consider those points corresponding to iterations where the penalty parameter is not modified. From condition (2.16) on the linesearch and assumption A1, we have $F(x_k) \geq \beta_F > -\infty$ for all k. Also, from Lemma 2.1 $\|\lambda_k\|$ is bounded, implying that

$$L_A(x_k, \lambda_k, s_k, \rho_k) \geq \beta_F - \max\left(\frac{\beta_\mu^2}{2\beta_\rho}, m\beta_\mu\beta_c \right) > -\infty. \tag{3.18}$$

Since $\|x_k\|$ is bounded when $\rho_k \neq \rho_{k-1}$ and $L_A(x_k, \lambda_k, s_k, \rho_k)$ is *reduced* when $\rho_k = \rho_{k-1}$ it follows that $L_A(x_k, \lambda_k, s_k, \rho_k)$ is bounded. Moreover, the reduction in $L_A(x_k, \lambda_k, s_k, \rho_k)$ is bounded for a sequence of iterations for which ρ_k is not changed. Let I denote the index at which ρ_k is modified and let $I \leq k \leq K$ denote the iterates for which ρ_k remains fixed. It follows from the above reasoning that there exists N such that

$$\phi_I - \phi_K = \sum_{k=I}^{K} (\phi_k - \phi_{k+1}) \leq N, \tag{3.19}$$

where to simplify the notation we have used $\phi_k \equiv \phi_k(0)$.

From the termination condition for the linesearch (2.15a), (3.3) and (3.19), we also have

$$\tfrac{1}{2}\sigma\beta_{svH} \sum_{k=I}^{K} \alpha_k \|p_k\|^2 \leq \sum_{k=I}^{K} (\phi_k - \phi_{k+1}) \leq N. \tag{3.20}$$

This result implies that $\alpha_k \|p_k\|$ is bounded. Hence if $\|x_k\|$ is not bounded there must exist sets of iterates with indices, say $s_l + 1 \leq k \leq r_l$ for $l = 1, 2, \ldots$, such that $\|x_{s_l}\| \leq M$, $\|x_k\| > M$ for M large enough, $\lim_{l\to\infty} r_l = \infty$, and $\lim_{l\to\infty} \|x_{r_l}\| \to \infty$. It follows that if

M is chosen so that $M > \max\{\|x_I\|\}$ then ρ_k is constant in the interval $s_l \leq k \leq r_l$. The existence of an index such that $\|x_{s_l}\| \leq M$ is assured since we have $\|x_I\| \leq M$ and at least one index in the interval for which $\|x_k\| > M$. From these assumptions and definitions it follows that

$$\sum_{k=s_l}^{r_l-1} \alpha_k \|p_k\| \geq \|x_{r_l} - x_{s_l}\| \rightarrow \infty. \tag{3.21}$$

It follows from Lemma 3.2 that $\|p_k\| > \epsilon$ for $s_l + 1 \leq k \leq r_l$. From (3.21) we get

$$\sum_{j=s_l}^{r_l-1} \alpha_j \|p_j\|^2 > \epsilon \sum_{j=s_l+1}^{r_l-1} \alpha_j \|p_j\| + \alpha_{s_l} \|p_{s_l}\|^2 \rightarrow \infty,$$

but this contradicts (3.20), implying that the points generated by the algorithm must lie on a compact set. ∎

To complete this section, we show that the search direction computed from the QP subproblem is bounded.

Lemma 3.5. *Under the assumptions of Lemma 3.4, the sequence $\{p_k\}$ is bounded.*

Proof. We drop the subscript k in the proof.

As all the steps taken in the solution of the QP subproblem are descent steps, we have from (2.3),

$$\psi(p_0) \geq \psi(p) = g^T p + \tfrac{1}{2} p^T H p = \tfrac{1}{2} \|H^{\frac{1}{2}} p + H^{-\frac{1}{2}} g\|_2^2 - \tfrac{1}{2} g^T H^{-1} g,$$

implying from **HC2** and $\|a\| \leq \|a + b\| + \|b\|$,

$$\sqrt{\beta_{svH}} \|p\| \leq \|H^{\frac{1}{2}} p\| \leq \|H^{-\frac{1}{2}} g\| + \|H^{\frac{1}{2}} p + H^{-\frac{1}{2}} g\| \leq \|H^{-\frac{1}{2}} g\| + \sqrt{2\psi(p_0) + g^T H^{-1} g}.$$

The boundedness of $\|p\|$ follows from this result Lemma 3.4, conditions **HC1** and **HC2** and the bound (3.8). ∎

It is tacitly assumed in the remaining proofs that the assumptions **A1–A7** and conditions **MC1**, **MC2**, **HC1** and **HC2** hold.

The sequence of search directions $\{p_k\}$

In this section we relate the behavior of the sequence $\{x_k - x^*\}$, where x^* denotes a KKT point closest to x_k, to that of the sequence $\{p_k\}$. In particular, we show that $\|p_k\| \rightarrow 0$ implies $x_k \rightarrow x^*$, and so it is enough to prove that $\|p_k\| \rightarrow 0$ to establish global convergence.

Although the KKT point x^* introduced above may not be unique, the assumptions made on the problem, and more specifically assumption **A7**, imply that if $\|x_k - x^*\|$ is sufficiently small then x^* is unique, as the following Lemma shows. This result allows us to work with a well-defined sequence $\{x_k - x^*\}$, at least close to a KKT point; it will also imply that the limit point of the sequence generated by the algorithm is unique.

Lemma 3.6. *The KKT points for problem NP are isolated.*

Proof. Assume that the result does not hold, and let x^* denote a KKT point for NP that is not isolated, that is, for any $\epsilon > 0$ there exists a KKT point $y_\epsilon \neq x^*$ satisfying $\|x^* - y_\epsilon\| < \epsilon$. Consequently, there exists a sequence $\{y_k\}$ such that y_k is a KKT point for all k, $y_k \neq x^*$ and $y_k \to x^*$.

For sufficiently small $\|x^* - y_k\|$ the active sets at y_k and x^* must be the same; otherwise we would have for some constraint j that $c_j(x^*) = 0$ with both $c_j(y_k) > 0$ and $(\lambda_k)_j = 0$ along some subsequence, where λ_k is the multiplier vector at y_k. From assumptions **A3** and **A4** and (1.1) we have $\lambda_k \to \lambda^*$, the multiplier vector at x^*, but this would imply $c_j(x^*) = \lambda_j^* = 0$, contradicting assumption **A6**.

Let Z_k denote a basis for the null-space of $\nabla\hat{c}(y_k)$, the Jacobian of the active constraints at y_k, and Z^* denote the corresponding basis at x^*. Among all possible bases, Z_k is selected to have continuous first derivatives in a ball around x^*. It follows from **A4** and the fact the active set is constant that such bases exist.

For any element of the sequence y_k and for x^* we have from (1.1)

$$Z_k^T \nabla F(y_k) = 0 \qquad \text{and} \qquad Z^{*T} \nabla F(x^*) = 0.$$

The Taylor series expansion of $Z_k^T \nabla F(y_k)$ around x^* gives

$$
\begin{aligned}
0 = Z_k^T \nabla F(y_k) &= Z_k^T(\nabla F(y_k) - \nabla c(y_k)^T \lambda^*) \\
&= Z^{*T}(\nabla F(x^*) - \nabla c(x^*)^T \lambda^*) + (\nabla Z(x^*)(\nabla F(x^*) - \nabla c(x^*)^T \lambda^*) \\
&\quad + Z^{*T} \nabla^2 L(x^*, \lambda^*))(y_k - x^*) + o(\|y_k - x^*\|),
\end{aligned} \tag{3.22}
$$

where $L(x, \lambda)$ is the Lagrangian function of NP. Using (1.1) in (3.22), and dividing by $\|y_k - x^*\|$ gives

$$Z^{*T} \nabla^2 L(x^*, \lambda^*)\delta_k = o(1), \qquad \text{where} \qquad \delta_k = \frac{y_k - x^*}{\|y_k - x^*\|}. \tag{3.23}$$

Let $\hat{c}$ denote the subset of constraints active at x^* and y_k. If ϵ is sufficiently small then δ_k satisfies

$$\hat{c}(y_k) = 0 = \nabla\hat{c}(x^*)(y_k - x^*) + o(\|y_k - x^*\|) \quad \Rightarrow \quad \nabla\hat{c}(x^*)\delta_k = o(1). \tag{3.24}$$

Finally, for any convergent subsequence of the bounded sequence $\{\delta_k\}$, with limit δ, we have from (3.23) and (3.24),

$$Z^{*T} \nabla^2 L(x^*, \lambda^*)\delta = 0, \qquad \nabla\hat{c}(x^*)\delta = 0,$$

contradicting assumption **A7**. ∎

This result, together with **A2**, implies that the number of KKT points lying on any compact region is finite. The distinctness and finiteness of the KKT points implies the existence of $\epsilon^* > 0$ such that for any two KKT points, say x_1^* and x_2^*, we have $\|x_1^* - x_2^*\| > 2\epsilon^*$ It follows that if $\|x_k - x^*\| < \epsilon^*$, where x^* is a KKT point nearest to x_k, then x^* is unique.

We now analyze the sequence of search directions $\{p_k\}$. The following result shows that as $p_k \to 0$ we get close to KKT points of NP and we only need to consider values p_k obtained as the minimizers for the corresponding subproblems. We complete this result by showing that a small value of $\|p_k\|$ also implies that the correct active set at x^* is identified, in the sense that the active QP constraints at p_k correspond to the active NP constraints at x^*.

Lemma 3.7. *If along a subsequence $p_k \to 0$ then along this subsequence $\|x_k - x^*\| \to 0$, where x^* is a KKT point nearest to x_k. For k large enough, x^* is unique, p_k is the QP minimizer and the correct active set at x^* is identified.*

Proof. A subsequence such that $p_k \to 0$ exists if and only if a subsequence exists such that $p_k \to 0$ and the active set at p_k is constant. Let $\{r\}$ denote the sequence of indices for such a subsequence.

From the definition (2.21) of p_r it follows immediately that $A_r p_r + c_r \geq 0$. From $p_r \to 0$ and assumption **A3** it must hold that $c_r^- \to 0$ and $\tilde{p}_r \to 0$.

From (2.5) we have

$$A_r^T \pi_r - g_r - H_r \tilde{p}_r = 0 \quad \text{and} \quad \pi_r^T(A_r \tilde{p}_r + c_r) = 0. \tag{3.25}$$

Since $\tilde{p}_r \to 0$ it follows that

$$A_r^T \pi_r - g_r \to 0, \quad \pi_r^T c_r \to 0 \quad \text{and} \quad c_r^- \to 0. \tag{3.26}$$

We now show that for large enough r that p_r must have been computed as the minimizer for the QP. It follows from $p_r \to 0$ and $\|d_r\| = 1$ that either there exists K such that for all $r > K$ we have $\gamma_r = 0$ or $\gamma_r \to 0$ (see (2.24)). If we assume the latter it follows that

$$\min(\bar{\gamma}_r, \hat{\gamma}_r) \to 0.$$

- If $\bar{\gamma}_r \to 0$ along a subsequence, then from (2.25) along this subsequence we will have for some constraint j

$$\nabla c_j(x_r)^T(\tilde{p}_r + \bar{\gamma}_r d_r) + c_j(x_r) = 0 \quad \text{and} \quad (\pi_r)_j = 0,$$

which implies that

$$c_j(x_r) \to 0 \quad \text{and} \quad (\pi_r)_j = 0,$$

contradicting assumption **A6**.

- If $\hat{\gamma}_r \to 0$ along a subsequence, then from (2.22),

$$\frac{\psi_r'(0)}{d_r^T H_r d_r} \to 0,$$

which implies from condition **HC1** and $\|d_r\| = 1$ that $\psi_r'(0) = (H_r \tilde{p}_r + g_r)^T d_r \to 0$. This result and condition (2.22) on d imply that for some constraint j we have $(\pi_r)_j < 0$, $(\pi_r)_j \to 0$ and $\nabla c_j(x_r)^T \tilde{p}_r + c_j(x_r) = 0$, giving

$$c_j(x_r) \to 0 \quad \text{and} \quad (\pi_r)_j \to 0,$$

and again contradicting assumption **A6**.

We conclude therefore that $\gamma_r = 0$ for $r > K$ and this together with (3.25) implies p_r is the minimizer of the QP subproblem. For r large enough $\pi_r \geq 0$, which together with (3.26) and assumption **A3** implies $\|x_r - x^*\| \to 0$, where x^* is the nearest KKT point to x_r. For r large enough x^* is unique.

Finally, we prove that for r large enough the active set of the QP coincides with the active set of NP at x^*. First note that for r large enough the active set of the QP must be a subset of the constraints active at x^*, otherwise p_r is a step to a nonactive constraint implying $\|p_r\| > \epsilon > 0$. Assume that for the subsequence we have $\nabla c_j(x_r)p_r + c_j(x_r) > 0$ and $c_j(x^*) = 0$. From (2.5b) we must have $(\pi_r)_j = 0$, implying from the convergence of π_r that $\lambda_j^* = 0$, but this violates assumption **A6**, and for r large enough the correct active set is known. ∎

This result shows that there is an $\epsilon > 0$ such that if $\|p_k\| < \epsilon$, then p_k is the solution of the QP subproblem, and the correct active set is known.

We have just shown that if $p_k \to 0$ along a subsequence, then $x_k \to x^*$. To show $p_k \to 0$, we need a stronger result, giving a relationship between the rates of convergence of the sequences $\{x_k - x^*\}$ and $\{p_k\}$.

Lemma 3.8. *If x^* denotes a KKT point closest to x_k, then there exists a constant M such that*

$$\|x_k - x^*\| \leq M\|p_k\|.$$

Proof. If $\|p_k\| > \epsilon$ for all k then the result holds trivially since $\|x_k\|$ and $\|x^*\|$ are both bounded. Again let $\{r\}$ denote the indices of a subsequence such that $p_r \to 0$ and the active set at p_r is constant. From Lemma 3.7, for this subsequence we have $\|x_r - x^*\| \to 0$. We assume for the rest of this proof that r is large enough so that x^* is unique, p_r is the minimizer of the QP and the correct active set has been identified.

Let $\hat{c}$, $\hat{A}$ and $\hat{\pi}$ denote the corresponding quantities restricted to the constraints in the active set. From assumption **A4** we know that $\hat{A}^*$ has full row rank, and we assume that r is large enough so that $\hat{A}_r$ also has full rank.

Let Z_r denote a basis for the null space of $\hat{A}_r$, with uniformly bounded norm and continuous first derivatives. From the optimality conditions for p_r, (2.5), we get

$$h(x) \equiv \begin{pmatrix} Z_r^T H_r \\ \hat{A}_r \end{pmatrix} p_r = - \begin{pmatrix} Z_r^T g_r \\ \hat{c}_r \end{pmatrix} = - \begin{pmatrix} Z_r^T(g_r - \hat{A}_r^T \hat{\lambda}^*) \\ \hat{c}_r \end{pmatrix}. \tag{3.27}$$

Since $h(x^*) = 0$, we have from the Taylor series expansion that

$$h_j(x_r) = S_j((\theta_r)_j)(x_r - x^*),$$

where $S_j((\theta_r)_j) = \nabla h_j(x^* + (\theta_r)_j(x_r - x^*))$ and $0 < (\theta_r)_j \leq 1$. We have therefore

$$\begin{pmatrix} Z_r^T g_r \\ \hat{c}_r \end{pmatrix} = -S(\theta_r)(x_r - x^*). \tag{3.28}$$

From (3.22) we get

$$S(0) = \begin{pmatrix} Z^{*T} \nabla^2 L(x^*, \lambda^*) \\ \hat{A}(x^*) \end{pmatrix},$$

and assumptions **A4** and **A7** imply that $S(0)$ is non-singular. It follows that for sufficiently large values of r, $S(\theta_r)$ is also nonsingular. It then follows from (3.28) that for some positive constant M_1,

$$\|x_r - x^*\| \leq M_1(\|Z_r^T g_r\| + \|\hat{c}_r\|). \tag{3.29}$$

From assumption **A3**, property **HC1** and (3.27) it follows that

$$M_2 \|p_r\| \geq \|Z_r^T g_r\| + \|\hat{c}_r\|, \tag{3.30}$$

for some positive constant M_2.

Since the subsequence $\{p_k\}$ such that $p_k \to 0$ is composed of a finite number of subsequences for which $p_r \to 0$ and the active set at p_r is constant, the required result follows from (3.29) and (3.30). ∎

Bounds on the penalty parameter

The conditions we have imposed on the algorithm (and more specifically on the multiplier estimate) are not sufficient to ensure that the penalty parameter is bounded. However, bounds on ρ_k are related to the behavior of different quantities in the algorithm, and in particular to $\|p_k\|$ and $\|c_k - s_k\|$. The following Lemmas introduce bounds on the size of ρ_k in terms of these quantities. We start by presenting the results for those iterations where the penalty parameter is modified, and then we extend the results to general iterations.

The notation k_l is used in all that follows to indicate iterations at which the value of the penalty parameter needs to be modified.

Lemma 3.9. *For any iteration k_l in which the value of ρ is modified,*

$$\rho_{k_l} \|c_{k_l} - s_{k_l}\| \leq N \quad \text{and} \quad \rho_{k_l} \|p_{k_l}\|^2 \leq N,$$

for some constant N.

Proof. All quantities in the proof refer to iteration k_l, and so this subscript is dropped.
From the definition of $\hat{\rho}$, (2.13), and Lemma 3.3 we get

$$\hat{\rho} \|c - s\|^2 = g^T p + \tfrac{1}{2} p^T H p + (2\lambda - \mu)^T (c - s)$$
$$\leq -\beta_1 p^T H p + \beta_2 \|c - s\| + (2\lambda - \mu)^T (c - s) \leq (\beta_2 + \|2\lambda - \mu\|)\|c - s\|,$$

where β_1 and β_2 are positive constants. From (3.14) and the above result we obtain the first bound in the Lemma,

$$\hat{\rho} \|c - s\| \leq 3\beta_\mu + \beta_2. \tag{3.31}$$

If the penalty parameter needs to be modified, condition (2.11) cannot hold for $\bar{\rho} \equiv \rho_{k_l-1}$, and (3.2) implies

$$\phi'(0) = g^T p + (2\lambda - \mu)^T (c - s) - \bar{\rho} \|c - s\|^2 > -\tfrac{1}{2} p^T H p.$$

It follows that

$$g^T p + \tfrac{1}{2} p^T H p + (2\lambda - \mu)^T (c - s) > 0. \tag{3.32}$$

Replacing in (3.32) the bound for $g^T p + \tfrac{1}{2} p^T H p$ given in Lemma 3.3 we obtain

$$(2\lambda - \mu)^T (c - s) + \beta_2 \|c - s\| > \beta_1 p^T H p \quad \Rightarrow \quad \frac{3\beta_\mu}{\beta_1} \|c - s\| > p^T H p. \tag{3.33}$$

From condition **HC2** we have $\|p\|^2 \leq (1/\beta_{svH}) p^T H p$. If we multiply both sides of this inequality by $\hat{\rho}$ and use (3.33) to bound $p^T H p$, we obtain

$$\hat{\rho}\|p\|^2 \leq \hat{\rho}\frac{1}{\beta_{svH}}p^T H p \leq \frac{3\beta_\mu}{\beta_1\beta_{svH}}\hat{\rho}\|c - s\| \leq \frac{3\beta_\mu(3\beta_\mu + \beta_2)}{\beta_1\beta_{svH}},$$

where the last inequality follows from (3.31). The second desired bound then follows from $2\hat{\rho} \geq \rho$. ∎

We now extend these results to all iterations. To simplify notation, we shall use I and K to denote k_l and k_{l+1} respectively. Thus, the penalty parameter is increased at x_I and x_K in order to satisfy condition (2.11), and remains fixed at ρ_I for iterations $I, \ldots, K - 1$.

Lemma 3.10. *There exists a constant M such that for all l,*

$$\rho_{k_l} \sum_{k=k_l}^{k_{l+1}-1} \|\alpha_k p_k\|^2 < M. \tag{3.34}$$

Proof. For $I \leq k \leq K - 1$, property (2.15a) imposed by the choice of α_k, and the fact that the penalty parameter is not increased, imply that

$$\phi_k - \phi_{k+1} \geq -\sigma\alpha_k\phi_k'.$$

Summing these inequalities for $k = I$ to $K - 1$, $0 \leq \alpha_k \leq 1$ together with (3.3) gives

$$\tfrac{1}{2}\sigma\beta_{svH}\sum_{k=I}^{K-1}\|\alpha_k p_k\|^2 \leq \phi_I - \phi_K. \tag{3.35}$$

Consider the term $\rho_I(\phi_I - \phi_K)$. From (2.2),

$$\rho\phi = \rho F - \rho\lambda^T(c - s) + \tfrac{1}{2}\rho^2\|c - s\|^2.$$

This equation, together with the boundedness of $\rho_I\|c_I - s_I\|$ and $\rho_I\|c_K - s_K\|$ (implied by $\rho_K > \rho_I$ and Lemma 3.9), and that of the multiplier estimates (Lemma 2.1), implies that for some $M_1 > 0$,

$$\rho_I(\phi_I - \phi_K) \leq M_1 + \rho_I(F_I - F_K). \tag{3.36}$$

Consider now iterations for which $\|p_I\| \leq \epsilon$, so that Lemma 3.7 applies and p_I has been obtained as the minimizer for the subproblem (for all other iterations Lemma 3.9 implies that ρ_I is bounded, and the result follows from Assumption **A3**, (3.36) and (3.35)).

Expanding F_K and c_K about x_I, we get

$$F_K - F_I = (x_K - x_I)^T g_I + O(\|x_I - x_K\|^2) \tag{3.37a}$$

$$c_K - c_I = A_I(x_K - x_I) + O(\|x_I - x_K\|^2). \tag{3.37b}$$

From Lemma 3.8 we have

$$\|x_I - x^*\| \leq M_p\|p_I\| \quad \text{and} \quad \|x_K - x^*\| \leq M_p\|p_K\|. \tag{3.38}$$

As p_I was obtained as the solution of the QP, condition (2.5a) must hold with multiplier vector $\pi_I \geq 0$. This condition together with (3.37a), (3.37b) and (3.38) implies

$$F_I - F_K = (c_I - c_K)^T \pi_I + O\left(\max(\|p_I\|^2, \|p_K\|^2)\right). \tag{3.39}$$

Using again (2.5),

$$c_I{}^T \pi_I = -p_I{}^T A_I{}^T \pi_I = -g_I{}^T p_I - p_I{}^T H_I p_I.$$

Since ρ is increased at iteration I, we must have that condition (2.11) cannot hold at that iteration, implying

$$\phi'_I(0) = g_I{}^T p_I + (2\lambda_I - \mu_I)^T(c_I - s_I) - \rho_{I-1}\|c_I - s_I\|^2 > -\tfrac{1}{2} p_I{}^T H_I p_I.$$

The previous two results imply

$$\rho_I \pi_I{}^T c_I < -\rho_I \tfrac{1}{2} p_I{}^T H_I p_I + \rho_I(2\lambda_I - \mu_I)^T(c_I - s_I) - \rho_I \rho_{I-1}\|c_I - s_I\|^2,$$

and this, together with the positive-definiteness of H_I (condition **HC2**), the boundedness of the multipliers (condition **MC1** and Lemma 2.1) and Lemma 3.9, gives

$$\rho_I c_I{}^T \pi_I < \rho_I(2\lambda_I - \mu_I)^T(c_I - s_I) \leq M_2, \tag{3.40}$$

for some $M_2 > 0$.

Consider now the term $c_K{}^T \pi_I$ in (3.39). From $\pi_I \geq 0$ we must have

$$-\rho_I c_K^T \pi_I \leq \rho_I c_K^{-T} \pi_I$$

and from (2.9) we have $\|c_K^-\| \leq \|c_K - s_K\|$. Using $\rho_I < \rho_K$ and Lemma 3.9, we conclude that there exists a constant M_3 such that

$$-\rho_I c_K{}^T \pi_I < M_3. \tag{3.41}$$

Finally, consider the third term on the right-hand side of (3.39). It follows from Lemma 3.9 and the relation $\rho_I < \rho_K$ that there exists M_4 and M_5 such that

$$\rho_I \|p_I\|^2 < M_4 \quad \text{and} \quad \rho_I \|p_K\|^2 < M_5,$$

and hence for some constant M_6

$$\rho_I O\left(\max(\|p_I\|^2, \|p_K\|^2)\right) < M_6. \tag{3.42}$$

Combining (3.40), (3.41) and (3.42), we obtain the bound

$$\rho_I(F_I - F_K) < M_2 + M_3 + M_6.$$

which, together with (3.36) and (3.35) implies the desired result ∎

Lemma 3.11. *There exists a constant M such that, for all k,*

$$\rho_k \|c_k - s_k\| \leq M. \tag{3.43}$$

Proof. As in the preceding Lemma, let $I = k_l$ and $K = k_{l+1}$. From Lemma 3.9, (3.43) is immediate for $k = I$ and $k = K$.

To verify a bound for $k = I + 1, \ldots, K - 1$ we analyze some intermediate iterations k and $k + 1$. We drop the iteration subscript; also let quantities evaluated at x_{k+1} be denoted with a tilde.

From (2.8), $\rho_I(\tilde{c}_j - \tilde{s}_j) = \min(\rho_I \tilde{c}_j, \tilde{\lambda}_j)$. Consider the following two cases:

- If $\rho_I \tilde{c}_j \geq -|\tilde{\lambda}_j|$, then

$$\rho_I |\tilde{c}_j - \tilde{s}_j| \leq |\tilde{\lambda}_j|. \tag{3.44}$$

- Assume now that $\rho_I \tilde{c}_j < -|\tilde{\lambda}_j|$. Expanding the j-th constraint function around x_k gives

$$\tilde{c}_j = c_j + \alpha a_j^T p + O(\|\alpha p\|^2).$$

Rewriting the previous expression, we obtain:

$$\tilde{c}_j = (1 - \alpha)c_j + \alpha(a_j^T p + c_j) + O(\|\alpha p\|^2). \tag{3.45}$$

Adding and subtracting $(1 - \alpha)s_j$ on the right-hand side of (3.45) gives

$$\tilde{c}_j = (1 - \alpha)(c_j - s_j) + (1 - \alpha)s_j + \alpha(a_j^T p + c_j) + O(\|\alpha p\|^2). \tag{3.46}$$

Since s_j, $a_j^T p + c_j$, α and $1 - \alpha$ are all non-negative, we get

$$(1 - \alpha)s_j + \alpha(a_j^T p + c_j) \geq 0,$$

and using this bound in (3.46) we obtain

$$\tilde{c}_j \geq (1 - \alpha)(c_j - s_j) + O(\|\alpha p\|^2). \tag{3.47}$$

Since we assume $\rho_I \tilde{c}_j < -|\tilde{\lambda}_j|$ we have $\tilde{c}_j = \tilde{c}_j - \tilde{s}_j \leq 0$. Using this bound and $1 - \alpha \leq 1$ in (3.47) we get the following inequality:

$$-\tilde{c}_j = |\tilde{c}_j| = |\tilde{c}_j - \tilde{s}_j| \leq -(1 - \alpha)(c_j - s_j) + O(\|\alpha p\|^2) \leq |c_j - s_j| + O(\|\alpha p\|^2).$$

Multiplying both sides by ρ_I gives

$$\rho_I |\tilde{c}_j - \tilde{s}_j| \leq \rho_I |c_j - s_j| + \rho_I O(\|\alpha p\|^2). \tag{3.48}$$

For a given iteration $k \leq K - 1$ and constraint j we have one of the following two situations:

- For some iteration l, $I < l \leq k$, $\rho_I(c_l)_j \geq -|(\lambda_l)_j|$. If we add (3.48) for iterations $r = l, \ldots, k - 1$, and use (3.44), we get

$$\rho_I |(c_k)_j - (s_k)_j| \leq \rho_I |(c_l)_j - (s_l)_j| + \rho_I O(\sum_{r=l}^{k-1} \|\alpha_r p_r\|^2) \leq |(\lambda_l)_j| + \rho_I O(\sum_{r=l}^{k-1} \|\alpha_r p_r\|^2).$$

The boundedness of $\rho_I |(c_k)_j - (s_k)_j|$ then follows from Lemmas 2.1 and 3.10.

- For all iterations l, $I < l \leq k$ we have $\rho_I(c_l)_j < -|(\lambda_l)_j|$. We add (3.48) for $r = I$ to $k - 1$, to obtain

$$\rho_I |(c_k)_j - (s_k)_j| \leq \rho_I |(c_I)_j - (s_I)_j| + \rho_I O(\sum_{r=I}^{k-1} \|\alpha_r p_r\|^2),$$

and now the desired result follows from Lemmas 3.9 and 3.10. ∎

Boundedness of α_k

Given the result of Lemma 3.10, all that is left to establish the global convergence of the algorithm is to show that the steplength is bounded away from zero. As a consequence of the weak assumptions imposed on the multiplier estimate μ_k, it is not possible to show that such a bound exists. However, it can be proved that the bound does exist if there is no subsequence along which $\|p_k\| \to 0$. This is enough to prove convergence.

We first derive a bound on the norm of the second derivative along the linesearch.

Lemma 3.12. *For $0 \le \theta \le \alpha_k$, there exists a positive constant N such that*

$$\phi_k''(\theta) \le N.$$

Proof. We again drop the subscript k. From (3.1),

$$\nabla^2 L_A = \begin{pmatrix} \nabla^2 F - \sum_j (\lambda_j - \rho(c_j - s_j))\nabla^2 c_j + \rho A^T A & -A^T & -\rho A^T \\ -A & 0 & I \\ -\rho A & I & \rho I \end{pmatrix}.$$

From the definition of ϕ, given in (2.2), we get

$$\begin{aligned}
\phi''(\theta) = {} & p^T W p + \sum_j \rho(c_j(\theta) - s_j(\theta)) p^T \nabla^2 c_j(\theta) p \\
& + \rho(A(\theta)p - q)^T (A(\theta)p - q) - 2\xi^T (A(\theta)p - q),
\end{aligned} \tag{3.49}$$

where the argument θ denotes quantities evaluated at $x + \theta p$, except for $s(\theta) \equiv s + \theta q$ and

$$W \equiv \nabla^2 F(\theta) - \sum_j (\lambda_j + \theta\xi_j)\nabla^2 c_j(\theta).$$

We now derive bounds on the terms on the right-hand side of (3.49). For the first term we can write

$$p^T W p \le N_1 \|p^2\| \le M_1, \tag{3.50}$$

for some constant M_1, using assumption **A3**, the boundedness of $\|\lambda\|$ and $\|\xi\|$ (condition **MC1** and Lemma 2.1), and the boundedness of $\|p\|$ (Lemma 3.5).

Expanding c_j in a Taylor series about x gives

$$c_j(\theta) = c_j(x) + \theta a_j(x)^T p + \tfrac{1}{2}\theta^2 p^T \nabla^2 c_j(\theta_j)p,$$

where $0 < \theta_j < \theta$. Using (2.10) and multiplying both sides by ρ gives

$$\rho(c_j(\theta) - (s_j(\theta))) = \rho(1 - \theta)(c_j(x) - s_j) + \rho\tfrac{1}{2}\theta^2 p^T \nabla^2 c_j(\theta_j)p.$$

Lemma 3.11 implies that $\rho|c_j(x) - s_j|$ is bounded, Lemma 3.10 implies that $\rho\|\theta p\|^2$ is bounded for $\theta \le \alpha$, and assumption **A3** implies that $\|\nabla^2 c_j(\theta_j)\|$ is also bounded. Consequently,

$$\rho\,|(c_j(\theta) - s_j(\theta))| \le N,$$

where N is a constant. This result and Lemma 3.5 imply the second term in (3.49) is also bounded, that is,

$$\sum_j \left| \rho(c_j(\theta) - s_j(\theta))p^T \nabla^2 c_j(\theta)p \right| \le N_2 \|p\|^2 \le M_2, \tag{3.51}$$

where N_2 and M_2 are constants.

Consider now $\rho\|A(\theta)p - q\|^2$, the third term on the right-hand side of (3.49). Using Taylor series, we have

$$a_j(x + \theta p)^T p = a_j^T p + \theta p^T \nabla^2 c_j(\bar{\theta}_j) p, \qquad (3.52)$$

where $0 < \bar{\theta}_j < \theta$. From (2.10) and Lemmas 3.10 and 3.11, we obtain

$$\rho\|A(\theta)p - q\|^2 \leq M_3, \qquad (3.53)$$

where M_3 is a constant.

From (3.52), (2.10), assumption **A3** and the boundedness of $\|\xi\|$ (Lemma 2.1), the final term on the right-hand side of (3.49) is also bounded,

$$-2\xi^T(A(\theta)p - q) = -2\xi^T(Ap - q) + \sum_j \xi_j \theta p^T \nabla^2 c_j(\bar{\theta}_j) p \leq 2\xi^T(c - s) + N_4\|p\|^2 \leq M_4, \quad (3.54)$$

where N_4 and M_4 are constants.

The desired bound follows from (3.49), (3.50), (3.51), (3.53) and (3.54). ∎

Lemma 3.13. *For any $\epsilon > 0$, if $\|p_k\| \geq \epsilon$ there exists a value $\bar{\alpha}(\epsilon)$ such that $\alpha_k \geq \bar{\alpha}(\epsilon) > 0$, where α_k is the steplength computed by the algorithm.*

Proof. We drop the subscript k corresponding to the iteration number. We start by proving that $\hat{\alpha}$ (as defined in (2.14) and (2.15)) is bounded away from zero if $\|p\| > \epsilon$. If condition (2.14) is satisfied at a given iteration, then $\hat{\alpha} = 1$, trivially bounded away from zero. We assume therefore that $\hat{\alpha}$ is chosen to satisfy (2.15).

In the proof of Lemma 3.1 it was shown that the linesearch procedure was well defined, and in particular, that there exists a value $\hat{\alpha} \in (0, 1]$ satisfying (2.15) and such that condition (2.15b) is not satisfied for any value of $\alpha \in [0, \hat{\alpha})$; see (3.5), (3.7) and (3.6).

From the Taylor series expansion of ϕ' at $\hat{\alpha}$ we have

$$\phi'(\hat{\alpha}) = \phi'(0) + \hat{\alpha}\phi''(\theta),$$

where $0 < \theta < \hat{\alpha}$. Therefore, using (3.5) and noting that $\eta < 1$ and $\phi'(0) < 0$, we obtain

$$\hat{\alpha} = \frac{\phi'(\hat{\alpha}) - \phi'(0)}{\phi''(\theta)} = (1 - \eta)\frac{|\phi'(0)|}{\phi''(\theta)}. \qquad (3.55)$$

(Since $\hat{\alpha} > 0$, θ must be such that $\phi''(\theta) > 0$.)

If $\|p\| \geq \epsilon$, then from (3.3) we have that $|\phi'(0)| \geq \frac{1}{2}\beta_{svH}\epsilon^2$, and from Lemma 3.12 we also have $\phi''(\theta) \leq N$, implying

$$\hat{\alpha} \geq \frac{\beta_{svH}}{2N}\epsilon^2.$$

If condition (2.16) is satisfied for $\hat{\alpha}$, then the previous bound holds for α. Otherwise, for some constraint j we must have $h_j(\hat{\alpha}) \equiv c_j(x + \hat{\alpha}p) + \beta_c < 0$ (using the notation introduced in Lemma 3.1). If $h_j(0) \geq \frac{1}{2}\beta_c > 0$, from the continuity of h there exists a value $\tilde{\alpha} < \hat{\alpha}$ such that $h_j(\tilde{\alpha}) = 0$ and $h_j(\alpha) \geq 0$ for all $\alpha \in [0, \tilde{\alpha}]$. From the mean-value theorem

$$\tilde{\alpha} = \frac{h_j(\tilde{\alpha}) - h_j(0)}{h_j'(\theta)} = \frac{h_j(0)}{|h_j'(\theta)|},$$

for some $\theta \in [0, \tilde{\alpha}]$. But as $|h'_j(\theta)| = |a_j(x + \theta p)^T p| \leq K$ for some $K > 0$ (from assumption **A3** and the boundedness of $\|p\|$, Lemma 3.5), we have

$$\tilde{\alpha} \geq \frac{\beta_c}{2K}. \tag{3.56}$$

If $h_j(0) \leq \frac{1}{2}\beta_c$, we must have from (2.4b),

$$h'_j(0) = a_j^T p \geq -c_j = \beta_c - h_j(0) \geq \tfrac{1}{2}\beta_c.$$

From $h_j(0) \geq 0$ and $h_j(\hat{\alpha}) < 0$ there must exist a value $\dot{\alpha} < \hat{\alpha}$ such that $h'_j(\dot{\alpha}) < 0$, implying the existence of $\tilde{\alpha} < \dot{\alpha}$ such that $h'_j(\tilde{\alpha}) = 0$ and $h'_j(\alpha) \geq 0$ for all $\alpha \in [0, \tilde{\alpha}]$ (also, $h_j(\alpha) \geq 0$ for all $\alpha \in [0, \tilde{\alpha}]$). From the mean-value theorem,

$$\tilde{\alpha} = \frac{h'_j(\tilde{\alpha}) - h'_j(0)}{h''_j(\theta)} = \frac{h'_j(0)}{|h''_j(\theta)|},$$

for some $\theta \in [0, \tilde{\alpha}]$. But $h'_j(0) \geq \frac{1}{2}\beta_c$, and $|h''_j(\theta)| = |p^T \nabla^2 c_j(x + \theta p)p| \leq \bar{K}$ for some $\bar{K} > 0$, from assumption **A3** and the boundedness of $\|p\|$, Lemma 3.5, implying again

$$\tilde{\alpha} \geq \frac{\beta_c}{2\bar{K}}. \tag{3.57}$$

The procedure to construct α will ensure that $\alpha \geq \frac{1}{2}\tilde{\alpha}$, and so the result presented in the Lemma will hold. ∎

We can now prove the global convergence theorem for the algorithm.

Theorem 3.1. *The sequence $\{x_k\}$ generated by the algorithm converges to a unique KKT point for NP.*

Proof.

It follows from Lemma 3.8 that to prove $\|x_k - x^*\| \rightarrow 0$, it is sufficient to show

$$\lim_{k \rightarrow \infty} \|p_k\| \rightarrow 0. \tag{3.58}$$

If (3.58) is true then there exists K such that $\|x_k - x^*\| < \epsilon^*/2$ and $\|p_k\| < \epsilon^*$ for all $k > K$, where $2\epsilon^*$ is the minimum distance between two KKT points. It follows that x^* is unique for $k > K$ (the sequence converges to the unique KKT point nearest to x_K), otherwise it implies that for some $k > K$ that either $\|x_k - x^*\| > \epsilon^*/2$ or $\|p_k\| > \epsilon^*$. Consequently, to prove the theorem it is sufficient to show (3.58) is true.

If $\|p_k\| = 0$ for any k, the algorithm terminates and the theorem is true. Hence we assume that $\|p_k\| \neq 0$ for any k. If $p_k \nrightarrow 0$, there must exist a subsequence $\{p_l\}$, and a positive constant ϵ, such that $\|p_l\| > \epsilon$ for all l. In this case, from Lemma 3.13 there will exist a uniform lower bound on α_l, $\alpha_l \geq \bar{\alpha} > 0$, but then

$$\rho_l \|\alpha_l p_l\| \geq \bar{\alpha} \epsilon \rho_l \rightarrow \infty,$$

contradicting the fact that $\rho_k \|\alpha_k p_k\|$ is bounded (Lemma 3.10).

In the bounded case, we know that there exists a value $\bar{\rho}$ and an iteration index $\tilde{K}$ such that $\rho = \bar{\rho}$ for all $k \geq \tilde{K}$. Again, the proof is by contradiction. Consider only indices l such that $l > \tilde{K}$. Every such iteration after $\tilde{K}$ must yield a strict decrease in the merit function because the termination condition for the linesearch (2.15a), together with the boundedness of the steplength (from Lemma 3.13 and $\|p_l\| > \epsilon$) and (3.3) imply

$$\phi_l(\alpha_l) - \phi_l(0) \leq \sigma\alpha_l\phi_l'(0) \leq -\tfrac{1}{2}\sigma\bar{\alpha}\beta_{svH}\|p_l\|^2 < 0.$$

The adjustment of the slack variables s in (2.7) can only lead to a further reduction in the merit function, as L_A is quadratic in s and the minimizer with respect to s_j is given by $c_j - \lambda_j/\rho$. From the fact that the penalty parameter is not modified, for iterations from the subsequence we have

$$\phi(x_{l+1}) - \phi(x_l) \leq -\tfrac{1}{2}\sigma\bar{\alpha}\beta_{svH}\epsilon^2.$$

Therefore, since the merit function with $\rho = \bar{\rho}$ decreases by at least a fixed quantity at every step in the subsequence, it must be unbounded below, contradicting (3.18). It follows that (3.58) must hold. ∎

Having established the global convergence of the algorithm, the next step is to show that the multiplier estimate $\lambda_k \to \lambda^*$. In order to prove this result, we need to strengthen our conditions on the multiplier estimate μ_k (if μ_k does not converge then λ_k will not converge either). The additional condition is

MC3. $\|\mu_k - \lambda^*\| = O(\|x_k - x^*\|)$, where λ^* denotes any multiplier vector associated with a KKT point closest to x_k.

This condition requires that β_μ in condition **MC1** be chosen so that

$$\beta_\mu \geq \|\lambda^*\|. \tag{3.59}$$

Estimates satisfying **MC1, MC2** and **MC3** may be obtained by computing a multiplier for the "active" constraints (say, least-squares estimates of least length), and expanding to the full multiplier space by augmenting this vector with zeros corresponding to the inactive constraints. If such an estimate does not satisfy **MC1**, then a suitable estimate may be determined by appropriate scaling. The multipliers at the stationary point of the QP also satisfy the three conditions. Note that if x^* is not unique then from Lemma 3.6, $\|x_k - x^*\| > \epsilon$ for some $\epsilon > 0$, and **MC3** holds for any vector μ_k that is bounded.

We first show that under the stronger conditions on μ_k the steplength α_k is uniformly bounded away from zero.

Lemma 3.14. *Under* **MC3** *and all earlier assumptions and conditions,* $\alpha_k \geq \bar{\alpha} > 0$.

Proof. We again drop the subscript k. We first tighten the bound on $\phi''(\theta)$ given in Lemma 3.12. From (3.50) and (3.51), we have that the first two terms on the right-hand side of (3.49) are bounded by a multiple of $\|p\|^2$. From (3.52), (2.10), (3.54) and Lemmas 3.10 and 3.11 we may obtain the following bound on the remaining terms on the right-hand side of (3.49)

$$\rho\,(A(\theta)p - q)^T(A(\theta)p - q) - 2\xi^T(A(\theta)p - q) \leq \rho\,(c - s)^T(c - s) + 2\xi^T(c - s) + M\|p\|^2, \tag{3.60}$$

for some constant M.

Observe that from (3.2) and (2.4b),

$$\rho(c-s)^T(c-s) + 2\xi^T(c-s) = -\phi'(0) + p^T g + \mu^T(c-s)$$
$$= -\phi'(0) + p^T(g - A^T\mu) - \mu^T s. \qquad (3.61)$$

Using Taylor expansions and Lemma 3.8 it follows that

$$p^T(g - A^T\mu) = p^T(g^* - {A^*}^T\mu) + O(\|p\|^2) = (\lambda^* - \mu)^T A^* p + O(\|p\|^2).$$

From this result and **MC3** there exists a constant $\tilde{M}$ such that

$$p^T(g - A^T\mu) \leq \tilde{M}\|p\|^2. \qquad (3.62)$$

From $\mu_k \to \lambda^*$, strict complementarity at a KKT point (assumption **A6**), and the fact that the correct active set is identified for $\|p\|$ small enough (Lemma 3.7), we eventually have $\mu \geq 0$ and $\mu^T s \geq 0$. Consequently, it follows from (3.49), (3.50), (3.51), (3.60), (3.61) and (3.62) that

$$\phi_k''(\theta) \leq -\phi_k'(0) + N\|p_k\|^2$$

for some constant $N > 0$. This result and (3.3) can be used with (3.55) to imply that there exists a value $\hat{\alpha}$ satisfying (2.15) such that

$$\hat{\alpha} \geq (1-\eta)\frac{\beta_{svH}\|p^2\|}{(\beta_{svH} + 2N)\|p^2\|} = (1-\eta)\frac{\beta_{svH}}{(\beta_{svH} + 2N)} > 0,$$

The desired result then follows from an argument identical to that given in the final part of Lemma 3.13. ∎

This lemma also implies that the effort needed to compute the value for the steplength is uniformly bounded in the algorithm. We now establish the convergence of the multiplier estimate.

Theorem 3.2. *Under* **MC3** *and all other assumptions and conditions,*

$$\lim_{k\to\infty} \lambda_k = \lambda^*.$$

Proof. From (2.27),

$$\lambda_{k+1} = \sum_{j=0}^{k} \gamma_{jk}\mu_j, \qquad (3.63)$$

where

$$\gamma_{kk} = \alpha_k', \qquad \gamma_{lk} = \alpha_l' \prod_{r=l+1}^{k}(1 - \alpha_r'), \quad l < k, \qquad (3.64)$$

with $\alpha_0' = 1$ and $\alpha_l' = \alpha_l$, $l \geq 1$. (This convention is used because of the special initial condition that $\lambda_0 = \mu_0$.) From Lemma 3.14 and (3.64), we observe that

$$0 < \bar{\alpha} \leq \alpha_l' \leq 1 \quad \text{for all } l, \qquad (3.65a)$$

$$\sum_{l=0}^{k} \gamma_{lk} = 1, \qquad (3.65b)$$

$$\gamma_{lk} \leq (1 - \bar{\alpha})^{k-l}, \quad l < k. \qquad (3.65c)$$

From condition **MC3** we have

$$\mu_k = \lambda^* + M_k \delta_k t_k, \tag{3.66}$$

with $|M_k| \leq M$, $\delta_k = \|x_k - x^*\|$ and $\|t_k\| = 1$. From Theorem 3.1, for any $\epsilon > 0$ we can choose a value K_1 so that, for $k \geq K_1$,

$$|M_k \delta_k| \leq \tfrac{1}{2}\epsilon. \tag{3.67}$$

Given any $\epsilon > 0$, we can also define an iteration index K_2 with the following property:

$$(1 - \bar{\alpha})^k \leq \frac{\epsilon}{2(k+1)(1+2\beta_\mu)} \tag{3.68}$$

for $k \geq K_2 + 1$. Let $K = \max(K_1, K_2)$. Then, from (3.63) and (3.66), we have for $k \geq 2K$,

$$\lambda_{k+1} = \sum_{l=0}^{K} \gamma_{lk} \mu_l + \sum_{l=K+1}^{k} \gamma_{lk}(\lambda^* + M_l \delta_l t_l).$$

Hence it follows from (3.65b) that

$$\lambda_{k+1} - \lambda^* = \sum_{l=0}^{K} \gamma_{lk}(\mu_l - \lambda^*) + \sum_{l=K+1}^{k} \gamma_{lk} M_l \delta_l t_l.$$

From the bounds on $\|\mu_l\|$ (condition **MC1**), $\|t_l\|$, and (3.59), we obtain

$$\|\lambda_{k+1} - \lambda^*\| \leq 2\beta_\mu \sum_{l=0}^{K} \gamma_{lk} + \sum_{l=K+1}^{k} \gamma_{lk} |M_l \delta_l|. \tag{3.69}$$

Since we assume $k \geq 2K$, it follows from (3.65a) and (3.65c) that

$$\sum_{l=0}^{K} \gamma_{lk} \leq \sum_{l=0}^{K} (1 - \bar{\alpha})^{k-l} \leq \sum_{l=0}^{K} (1 - \bar{\alpha})^{2K-l} \leq (K+1)(1 - \bar{\alpha})^K.$$

Using (3.68), we thus obtain the following bound for the first term on the right-hand side of (3.69):

$$2\beta_\mu \sum_{l=0}^{K} \gamma_{lk} \leq \tfrac{1}{2}\epsilon. \tag{3.70}$$

To bound the second term in (3.69), we use (3.65b) and (3.67):

$$\sum_{l=K+1}^{k} \gamma_{lk} |M_l \delta_l| \leq \tfrac{1}{2}\epsilon \sum_{l=K+1}^{k} \gamma_{lk} \leq \tfrac{1}{2}\epsilon. \tag{3.71}$$

Combining (3.69)–(3.71), we obtain the following result: given any $\epsilon > 0$, we can find K such that

$$\|\lambda_k - \lambda^*\| \leq \epsilon \quad \text{for} \quad k \geq 2K + 1,$$

which implies the desired result. ∎

4. Rate of convergence

In this Section we shall show under additional assumptions on the multiplier estimate that the algorithm converges at a superlinear rate, independently of the asymptotic behavior of the penalty parameter.

Since $p_k \to 0$, we may assume without loss of generality that p_k has been obtained as the minimizer for the QP subproblem, and that the correct active set has been identified. We again start by presenting an outline of the steps taken.

- Bounds on the rate of growth of the penalty parameter introduced in Lemmas 3.9, 3.10 and 3.11 are tightened.

  - In Lemma 4.1 we prove that al ! iterations at which ρ_k is increased (if we have an infinite sequence of such iterations)
    $$\rho_k \|c_k - s_k\| \to 0 \quad \text{and} \quad \rho_k \|p_k\|^2 \to 0.$$

  - In Lemma 4.2 and Lemma 4.3 these results are extended to all iterations.

- In Lemma 4.4 it is shown that $\mu_k^T s_k = 0$ for sufficiently large k.

- Lemma 4.5 proves the superlinear convergence of the sequence $\{x_k + p_k - x^*\}$, under certain assumptions on H_k.

- For k sufficiently large, $\alpha_k = 1$.

  - Lemma 4.6 gives the relationship between the descent in one iteration $\phi_k(1) - \phi_k(0)$ and the initial derivative in the linesearch $\phi_k'(0)$.

  - Theorem 4.1 shows that $\alpha_k = 1$ for all sufficiently large k, implying superlinear convergence.

- Finally, Theorem 4.2 shows that under an additional condition on the multipliers, the penalty parameter remains bounded.

The first two Lemmas introduce refinements on the results presented in Lemmas 3.9, 3.10 and 3.11, and their proofs are based on the corresponding proofs for these Lemmas.

Lemma 4.1. *If $k_l \to \infty$, where k_l denotes an iteration at which the penalty parameter is increased, then*
$$\lim_{l \to \infty} \rho_{k_l} \|c_{k_l} - s_{k_l}\| = 0 \quad \text{and} \quad \lim_{l \to \infty} \rho_{k_l} \|p_{k_l}\|^2 = 0.$$

Proof. We drop the subscript k_l in what follows.

Since p is the minimizer of QP, condition (2.5a) holds for a nonnegative vector π. From (2.4b) and (2.5a) we have $g^T p + \frac{1}{2} p^T H p = -\pi^T c$ and using this result in the definition of $\hat{\rho}$, (2.13),
$$\hat{\rho} \|c - s\|^2 = -\tfrac{1}{2} p^T H p + (2\lambda - \mu - \pi)^T (c - s) - \pi^T s \le \|2\lambda - \mu - \pi\| \|c - s\|.$$

From (2.12) we have $\rho \le 2\hat{\rho}$, and using Theorem 3.2, **MC3** and $\pi_{k_l} \to \lambda^*$ we obtain
$$\lim_{l \to \infty} \rho_{k_l} \|c_{k_l} - s_{k_l}\| \le 2 \lim_{l \to \infty} \|2\lambda_{k_l} - \mu_{k_l} - \pi_{k_l}\| = 0. \tag{4.1}$$

Finally, from (3.33) and (4.1) we have $\lim_{l \to \infty} \rho_{k_l} \|p_{k_l}\|^2 = 0$, completing the proof. ∎

Lemma 4.2. *For general iterations k, $\lim_{k\to\infty} \rho_k \|p_k\|^2 = 0$.*

Proof. Define $I \equiv k_l$ and $K \equiv k_{l+1}$.

If ρ is bounded, the result follows from Theorem 3.1. If ρ is increased in an infinite number of iterations, from (3.35) and Lemma 3.13 we only need to show that $\phi_I - \phi_K \to 0$.

From the boundedness of $\|\lambda_k\|$ (Lemma 2.1), Lemma 4.1 and the fact that $\rho_I < \rho_K$, we have

$$\rho_I |\lambda_I^T (c_I - s_I)| \leq 2\rho_I \|\lambda_I\| \, \|c_I - s_I\| \to 0,$$
$$\rho_I |\lambda_K^T (c_K - s_K)| \leq 2\rho_K \|\lambda_K\| \, \|c_K - s_K\| \to 0.$$

We also have from Lemma 4.1,

$$\rho_I^2 \|c_I - s_I\|^2 \to 0, \qquad \rho_I^2 \|c_K - s_K\|^2 \to 0.$$

These results and the definition of ϕ, (2.2), imply

$$\rho_I(\phi_I - \phi_K) - \rho_I(F_I - F_K) \to 0. \tag{4.2}$$

We now analyze the asymptotic behavior of the term $\rho_I(F_I - F_K)$. We have

$$F_I - F_K = (c_I - c_K)^T \pi_I + O\left(\max(\|p_I\|^2, \|p_K\|^2)\right).$$

Using the same arguments as in the proof of Lemma 3.10, inequality (3.40) also holds in this case, and from (3.14),

$$\rho_I \pi_I^T c_I < \rho_I \|c_I - s_I\| \|2\lambda_I - \mu_I\| \leq 3\beta_\mu \rho_I \|c_I - s_I\|. \tag{4.3}$$

A second bound for this term can be obtained from $\pi_I \geq 0$ and $s_I \geq 0$, implying

$$\rho_I \pi_I^T c_I \geq \rho_I \pi_I^T (c_I - s_I) \geq -\rho_I \|\pi_I\| \|c_I - s_I\|. \tag{4.4}$$

Since $\|\pi_I\|$ is bounded, it follows from applying Lemma 4.1 to (4.3) and (4.4) that

$$\rho_I \pi_I^T c_I \to 0. \tag{4.5}$$

From (2.9), the boundedness of $\|\pi_I\|$ and Lemma 4.1,

$$-\rho_I c_K^T \pi_I \leq \rho_I c_K^{-T} \pi_I \leq \rho_I \|\pi_I\| \|c_K - s_K\| \to 0. \tag{4.6}$$

We can again use Lemma 4.1 to obtain

$$\rho_I O\left(\max(\|p_I\|^2, \|p_K\|^2)\right) \to 0. \tag{4.7}$$

From (3.39), (4.5), (4.6) and (4.7) we have that the sequence $\{\rho_I(F_I - F_K)\}$ is bounded above by a sequence that converges to zero. It then follows from $\phi_I - \phi_K \geq 0$ and (4.2) that $\rho_I(\phi_I - \phi_K) \to 0$ and the desired result follows from (3.35) and Lemma 3.14. ∎

Lemma 4.3. *For general iterations k, $\lim_{k\to\infty} \rho_k \|c_k - s_k\| = 0$.*

Proof. If ρ is bounded the result follows from $c^* \geq 0$, $\lambda^* \geq 0$, $\lambda^{*T} c^* = 0$, Theorems 3.1 and 3.2 and (2.8).

We assume therefore that ρ is increased an infinite number of times. Consider two cases:

- If constraint j is such that $c_j^* > 0$, then $\lambda_j^* = 0$ and from (2.8),

$$\rho |c_j - s_j| = |\min(\rho c_j, \lambda_j)|,$$

  but from Theorem 3.2 and assumption **A6**, eventually $\lambda_j < \rho c_j$, implying

$$\rho |c_j - s_j| = |\lambda_j| \to 0.$$

- For those j such that $c_j^* = 0$, implying $\lambda_j^* > 0$, consider iteration indices large enough that the correct active set is identified (Lemma 3.7), implying $a_j^T p + c_j = 0$. From the Taylor series expansion for c_j and the boundedness of the steplength,

$$c_j(x_k + \alpha_k p_k) = c_j(x_k) + \alpha_k (a_k)_j^T p_k + O(\|\alpha_k p_k\|^2) = (1 - \alpha_k) c_j(x_k) + O(\|p_k\|^2),$$

Recurring this relationship for k, $I < k < K$, we get

$$\rho_k (c_k)_j = \rho_I (c_k)_j = \rho_I \prod_{l=I}^{k-1} (1 - \alpha_l)(c_I)_j + \rho_I O\Big(\sum_{l=I}^{k-1} \|p_l\|^2\Big),$$

but as $0 < \alpha_l \leq 1$ we must have

$$\rho_k |(c_k)_j| \leq \rho_I |(c_I)_j| + \rho_I O\Big(\sum_{l=I}^{k-1} \|p_l\|^2\Big). \tag{4.8}$$

From $c_j^* = 0$, assumption **A6** and (2.8), eventually it must hold that $\rho_I |(c_I)_j - (s_I)_j| = \rho_I |c(_I)_j|$, and using Lemma 4.1, (4.8) and Lemma 4.2,

$$\rho_k |(c_k)_j| \to 0.$$

From this result, definition (2.8), assumption **A6** and Theorem 3.2, for k large enough

$$\rho_k |(c_k)_j - (s_k)_j| = |\min(\rho_k (c_k)_j, (\lambda_k)_j)| = |\rho_k (c_k)_j| \to 0.$$

This completes the proof. ∎

Lemma 4.4. *For k large enough $\mu_k^T s_k = 0$.*

Proof. If constraint j is such that $c_j^* > 0$, then for k large enough $(c_k)_j \geq \epsilon > 0$, and $(a_k)_j^T p_k + (c_k)_j \geq \frac{1}{2}\epsilon > 0$. It therefore follows from **MC2** that $(\mu_k)_j = 0$.

If j is such that $c_j^* = 0$, then from assumption **A6**, $\lambda_j^* > 0$. Also, from Lemma 4.3, $\rho_k((c_k)_j - (s_k)_j) = \min(\rho_k (c_k)_j, (\lambda_k)_j) \to 0$, and for large enough k Theorem 3.2 will imply $\rho_k(c_k)_j \leq (\lambda_k)_j$; these two results and definition (2.7) imply

$$(s_k)_j = \max\Big(0, (c_k)_j - \frac{(\lambda_k)_j}{\rho_k}\Big) = 0,$$

completing the result. ∎

To prove that the algorithm converges superlinearly it is necessary to assume that H_k converges to an approximation of $\nabla^2_{xx}L(x^*, \lambda^*)$ in some sense, where $L(x, \lambda)$ denotes the Lagrangian function for problem NP.

Define W_k as

$$W_k \equiv \nabla^2_{xx}L(x_k, \lambda_k) = \nabla^2_{xx}F(x_k) - \sum_j (\lambda_k)_j \nabla^2_{xx}c_j(x_k). \tag{4.9}$$

We impose the following additional condition on H_k:

HC3. Following Boggs, Tolle and Wang [BTW82], we assume

$$\|Z_k^T(H_k - W_k)p_k\| = o(\|p_k\|),$$

where Z_k is a basis for the null space of $\hat{A}_k$, the Jacobian of x_k of those constraints active at x^*, that is bounded in norm and has its smallest singular value bounded away from 0.

The proof proceeds by first showing that the sequence $\{x_k + p_k - x^*\}$ converges superlinearly, and then proving that a steplength of one is eventually attained.

The following lemma corresponds to Theorem 3.1 in [BTW82].

Lemma 4.5. *Under assumptions* **A1–A7**, *and conditions* **MC1–MC3**, **HC1–HC3**,

$$\|x_k + p_k - x^*\| = o(\|x_k - x^*\|). \tag{4.10}$$

The results presented on bounds for the growth rate of the penalty parameter allow us to obtain an asymptotic expansion for the quantities involved in the linesearch termination criterion. We want to prove that condition (2.14) is satisfied for k sufficiently large. It is shown in the following lemma that the satisfaction of (2.14) is directly related to the asymptotic properties of $T_k \equiv p_k^T(g_k - A_k^T\mu_k) + p_k^T W_k p_k$.

Lemma 4.6. *The following relationship holds:*

$$\phi_k(1) - \phi_k(0) = \tfrac{1}{2}\phi_k'(0) + \tfrac{1}{2}T_k + o(\|p_k\|^2).$$

Proof. In the proof we drop the subscript k, and we denote quantities associated with $x_k + p_k$ by a tilde, that is, $\tilde{F} \equiv F(x_k + p_k)$ while $F \equiv F(x_k)$.

From the definition of the merit function (2.2) and (2.1) we have

$$\phi(1) - \phi(0) = \tilde{F} - F - \mu^T(\tilde{c} - s - q) + \lambda^T(c - s)$$
$$+ \frac{\rho}{2}(\tilde{c} - s - q)^T(\tilde{c} - s - q) - \frac{\rho}{2}(c - s)^T(c - s). \tag{4.11}$$

From the Taylor series expansion of c around x and (2.10) we have

$$\tilde{c}_j - s_j - q_j = \tilde{c}_j - c_j - a_j^T p = \tfrac{1}{2}p^T\nabla^2 c_j p + o(\|p\|^2),$$

and using this result with the Taylor expansions for c and F in (4.11) we obtain

$$\phi(1) - \phi(0) = g^T p + \tfrac{1}{2} p^T \nabla^2 F p - \tfrac{1}{2} \sum_j \mu_j \, p^T \nabla^2 c_j p + \lambda^T (c - s)$$
$$+ \tfrac{\rho}{8} \sum_j (p^T \nabla^2 c_j p)^2 - \tfrac{\rho}{2}(c - s)^T(c - s) + o(\|p\|^2). \tag{4.12}$$

From (2.6), condition **MC3** and Theorem 3.2 we have

$$\mu = \lambda + \xi = \lambda + o(1). \tag{4.13}$$

Also, from Lemma 4.2 and assumption **A3** we have $\rho p^T \nabla^2 c_j p = o(1)$. Replacing these results in (4.12) and reordering the terms we obtain

$$\phi(1) - \phi(0) = g^T p + \tfrac{1}{2} p^T \nabla^2 F p - \tfrac{1}{2}\sum_j \lambda_j \, p^T \nabla^2 c_j p + \tfrac{1}{2}(2\lambda - \mu)^T(c - s)$$
$$+ \tfrac{1}{2}\mu^T(c - s) - \tfrac{\rho}{2}(c - s)^T(c - s) + o(\|p\|^2).$$

Using (4.9) and (3.2) to simplify this expression,

$$\phi(1) - \quad , = \tfrac{1}{2}\phi'(0) + \tfrac{1}{2}\Big(g^T p + p^T W p + \mu^T(c - s)\Big) + o(\|p\|^2). \tag{4.14}$$

Finally, from condition **MC2** we have $\mu^T c = -\mu^T A p$, and from Lemma 4.4 we know that eventually $\mu^T s = 0$, implying in particular that $\mu^T s = o(\|p\|^2)$, and replacing these bounds in (4.14) we have

$$\phi(1) - \phi(0) = \tfrac{1}{2}\phi'(0) + \tfrac{1}{2}\Big(p^T W p + p^T(g - A^T \mu)\Big) + o(\|p\|^2),$$

completing the result. ∎

The main result of this section is given in the next theorem. It is shown that, if condition **MC3** is replaced by a stronger condition, then after a finite number of iterations a steplength of one is taken for all iterations thereafter, implying that the algorithm achieves superlinear convergence. The new condition is

MC3'. $\|\mu_k - \lambda^*\| = o(\|x_k - x^*\|)$.

It is possible to prove superlinear convergence without the need to strengthen the conditions on the multipliers. It is shown in [Pr89] that there exists a constant M such that if $\rho_k > M$, condition **MC3** is sufficient.

Theorem 4.1. *If* **MC3'** *and all other assumptions and conditions hold then eventually a unit step is always taken and the algorithm converges superlinearly.*

Proof. As in Powell and Yuan [PY86], observe that the continuity of second derivatives gives the following relationships:

$$
\begin{aligned}
F(x_k + p_k) &= F(x_k) + \tfrac{1}{2}\Big(g(x_k) + g(x_k + p_k)\Big)^T p_k + o(\|p_k\|^2) \\
c(x_k + p_k) &= c(x_k) + \tfrac{1}{2}\Big(A(x_k) + A(x_k + p_k)\Big)p_k + o(\|p_k\|^2).
\end{aligned}
\tag{4.15}
$$

From the Taylor series expansions we have

$$
\begin{aligned}
F(x_k + p_k) &= F(x_k) + g(x_k)^T p_k + \tfrac{1}{2} p_k^T \nabla^2 F(x_k) p_k + o(\|p_k\|^2) \\
c_j(x_k + p_k) &= c_j(x_k) + a_j(x_k)^T p_k + \tfrac{1}{2} p_k^T \nabla^2 c_j(x_k) p_k + o(\|p_k\|^2),
\end{aligned}
\tag{4.16}
$$

and since (4.10) and Lemma 3.8 imply $g(x_k+p_k) = g^* + o(\|p_k\|)$, $a_j(x_k+p_k) = a_j^* + o(\|p_k\|)$, we get from (4.15) and (4.16) that (we drop the subscript k)

$$
p^T \nabla^2 F p = (g^* - g)^T p + o(\|p\|^2) \tag{4.17a}
$$

$$
p^T \nabla^2 c_j p = (a_j^* - a_j)^T p + o(\|p\|^2). \tag{4.17b}
$$

Condition **MC3**, Theorem 3.2 and (4.13) give $\sum_j \lambda_j p^T \nabla^2 c_j p = \sum_j \mu_j \, p^T \nabla^2 c_j p + o(\|p\|^2)$, and if we apply this bound to the result of adding (4.17a) to (4.17b) multiplied by λ_j, we have

$$
p^T W p = p^T (g^* - A^{*T} \mu) - p^T (g - A^T \mu) + o(\|p\|^2). \tag{4.18}
$$

Condition **MC3'**, (1.1) and Lemma 3.8 imply

$$
p^T (g^* - A^{*T} \mu) = p^T A^{*T} (\lambda^* - \mu) = o(\|p\|^2),
$$

and from (4.18),

$$
T = p^T W p + p^T (g - A^T \mu) = p^T (g^* - A^{*T} \mu) + o(\|p\|^2) = o(\|p\|^2). \tag{4.19}
$$

From Lemma 4.6 and (4.19) we get

$$
\phi(1) - \phi(0) = \tfrac{1}{2} \phi'(0) + o(\|p\|^2).
$$

Since $\phi'(0) < 0$, the above relationship and Theorem 3.1 imply that condition (2.14) is eventually satisfied for k sufficiently large.

Regarding condition (2.16), we can use Taylor series expansions for c_j to write

$$
c_j(x_k + p_k) = c_j(x_k) + a_j(x_k + \theta_j p_k)^T p_k \tag{4.20}
$$

for some $\theta_j \in [0,1]$, and

$$
a_j(x_k + \theta_j p_k)^T p_k = a_j(x_k)^T p_k + p_k^T \nabla^2 c_j(x_k + \bar{\theta}_j p_k) p_k, \tag{4.21}
$$

for $\bar{\theta}_j \in [0, \theta_j]$.

Using Theorem 3.1 and the boundedness of $\|\nabla^2 c_j(x_k + \bar{\theta}_j p_k)\|$ (from Assumption **A3** and Lemma 3.4) in (4.21), for k large enough

$$
a_j(x_k + \theta_j p_k)^T p_k \geq a_j(x_k)^T p_k - \tfrac{1}{2} \beta_c,
$$

and from (2.4b),

$$
a_j(x_k + \theta_j p_k)^T p_k \geq a_j(x_k)^T p_k - \tfrac{1}{2} \beta_c \geq -c_j(x_k) - \tfrac{1}{2} \beta_c.
$$

Replacing this bound in (4.20), we obtain for all k large enough $c(x_k + p_k) \geq -\tfrac{1}{2}\beta_c\epsilon$, and condition (2.16) will also be satisfied, giving $x_{k+1} = x_k + p_k$. The required result then follows from Lemma 4.5. ∎

Boundedness of the penalty parameter

The last result in this section shows that, if condition **MC3'** is replaced by a slightly stronger condition, the penalty parameter needs to be modified in at most a finite number of iterations (and consequently it remains bounded). The criterion presented will be satisfied, for example, by the least-squares multipliers computed at $x_k + p_k$.

Theorem 4.2. *If the multiplier estimates μ_k in the algorithm satisfy*

$$\|\mu_k - \lambda^*\| = O(\|x_k + p_k - x^*\|), \tag{4.22}$$

and all other assumptions and conditions hold then there exists a constant M such that $\rho_k \leq M$ for all k.

Proof. We may assume k large enough so that $\alpha_k = 1$. From (2.5), (2.4b) and $\pi_k^T s_k \geq 0$, we have

$$g_k^T p_k + p_k^T H_k p_k = p_k^T A_k \pi_k = -c_k^T \pi_k \leq -(c_k - s_k)^T \pi_k, \tag{4.23}$$

where π_k denotes the QP multipliers at iteration k. From (3.2), (4.23) and the fact that a unit steplength is accepted, it follows that

$$\phi_k'(0) = \leq -p_k^T H_k p_k + \|2\mu_{k-1} - \mu_k - \pi_k\|\|c_k - s_k\| - \rho_k\|c_k - s_k\|^2. \tag{4.24}$$

From (4.22), **HC2**, Lemma 3.8 and $\|\pi_k - \lambda^*\| = O(\|p_k\|)$ we must have

$$\|2\mu_{k-1} - \mu_k - \pi_k\| \leq M_1 \|p_k\| \leq M_2 \sqrt{p_k^T H_k p_k}$$

for some positive constants M_1, M_2. It then follows using $a^2 + b^2 \geq 2ab$ that

$$\|2\mu_{k-1} - \mu_k - \pi_k\|\|c_k - s_k\| \leq M_2\sqrt{p_k^T H_k p_k}\|c_k - s_k\| \leq \tfrac{1}{2}p_k^T H_k p_k + \tfrac{1}{2}M_2^2\|c_k - s_k\|^2,$$

implying from (4.24) that

$$\phi_k'(0) \leq -\tfrac{1}{2}p_k^T H_k p_k + (\tfrac{1}{2}M_2^2 - \rho_k)\|c_k - s_k\|^2.$$

From this inequality it follows that if $\rho_k \geq \tfrac{1}{2}M_2^2$, condition (2.11) will be satisfied, and the penalty parameter will not be increased. Given that we are using the rule (2.12) for updating ρ_k, it must hold that $\rho_k \leq M_2^2$. ∎

5. Other Merit Functions

Several merit functions have been proposed and analyzed in the literature (a review can be found in Powell [Po87]). The question arises if the convergence results using early termination in the solution of the QP subproblem depend on our specific merit function, or if they are fairly independent of this choice. We shall show in this section that the choice of merit function is not critical. What we present is how to adapt our SQP algorithm to the use of other merit functions rather than examine other methods explicitly to see if the particular QP subproblem posed and the manner the search is performed can be adapted

to the use of an incomplete solution. For example, we still perform a search in the x and λ spaces. Slack variables do not appear in the merit functions we shall consider, consequently the search in the space of the slack variables is no longer required.

We have selected as examples the study of two particular merit functions. The first one corresponds to a class of merit functions that includes among others the ℓ_1 merit function analyzed in Han [Han76], Byrd and Nocedal [BN88] and Burke and Han [BH89]. This general merit function takes the form:

$$\phi(x, \lambda) = F(x) + \lambda^T c^-(x) + \rho \|c^-(x)\|_p, \tag{5.1}$$

where an ℓ_p norm $(1 \le p \le \infty)$ is used, and $c_j^-(x) \equiv \max(0, -c_j(x))$. The second merit function we consider is

$$\phi(x, \lambda) = F(x) + \lambda^T c^-(x) + \tfrac{1}{2}\rho \|c^-(x)\|_2^2. \tag{5.2}$$

This merit function has been studied among others by Powell and Yuan [PY86] (applied to the equality-constrained problems only) and Schittkowski [Sch81]. Unlike either of these algorithms, where the multiplier estimate λ was treated as a function of the iterate $\lambda(x)$, we do not explicitly define the form of the multiplier estimates although the ones used in both methods satisfy the criteria **MC1, MC2** and **MC3**. Indeed the one used in [PY86] also satisfies **MC3'**.

We still assume **A1–A7** hold for the problem. However, when the merit function (5.1) is used, the multiplier estimate μ_k is only required to satisfy **MC1**. This condition is trivial to satisfy. For example, we may choose $\lambda_0 = 0$ and $\mu_k = 0$ making the search in the multiplier space void. Such a choice reduces (5.1) to the well-known ℓ_1 merit function and our algorithm becomes very similar to that analyzed in [Han76]. When (5.2) is used, we assume conditions **MC1** and **MC2** hold. We have also assumed in the proofs that $\lambda_0 \ge 0$ and $\mu_k \ge 0$. We omit the proofs that the iterates lie on a compact set. For the first merit function (5.1) this proof is relatively straightforward, since it will be shown that the penalty parameter is bounded. The proof for the second merit function (5.2) is very similar to that for the Augmented Lagrangian merit function.

The criteria (2.15) for the choice of steplength α_k assume the merit function has continuous first derivatives. This property does not necessarily hold for the merit functions under consideration. Therefore we use the following criteria for determining a value α_k.

Define

$$\Delta_k \equiv g_k^T p_k + (\xi_k - \lambda_k)^T c^-(x_k) - \rho_k \|c^-(x_k)\|. \tag{5.3}$$

We start by selecting a value $\hat{\alpha}_k$ satisfying

$$\phi_k(\hat{\alpha}_k) \equiv \phi(x_k + \hat{\alpha}_k p_k, \lambda_k + \hat{\alpha}_k \xi_k) \le \phi_k(0) + \eta \hat{\alpha}_k \Delta_k, \tag{5.4}$$

and either

$$\hat{\alpha}_k \ge \gamma_l > 0 \tag{5.5}$$

or

$$\hat{\alpha}_k > \gamma_u \bar{\alpha}_k \quad \text{and} \quad \phi_k(\bar{\alpha}_k) > \phi_k(0) + \sigma \bar{\alpha}_k \Delta_k, \tag{5.6}$$

where $0 < \gamma_l < \gamma_u < 1$, $0 < \eta \le \sigma < 1$ and $\bar{\alpha}_k > 0$. For a discussion of these criteria and the existence of $\hat{\alpha}_k$ see Calamai and Moré [CM87].

In addition to these conditions, we also also want to limit the size of the infeasibilities. If $\hat{\alpha}_k$ satisfies condition (2.16), then we let $\alpha_k = \hat{\alpha}_k$. Otherwise, we compute α_k by performing a backtracking linesearch from $\hat{\alpha}_k$ until conditions (5.4) and (2.16) are both satisfied.

Our preference for the criteria given in Section 2 is based on our belief that in practice they lead to a better choice of α_k. In the definition of our algorithm we could have used other steplength criteria without impacting the convergence properties.

The following basic relationships will be used to establish the convergence results,

$$c_j^-(x + \alpha p) \le |c_j(x + \alpha p) - c_j(x) - \alpha a_j^T p| - \min(0, c_j(x) + \alpha a_j^T p) \quad (5.7a)$$

$$-\min(0, c_j(x) + \alpha a_j^T p) \le (1 - \alpha)c_j^-(x) \quad (5.7b)$$

$$-\omega^T A p \le -\|c^-(x)\|_l \quad (5.7c)$$

$$-\Omega A p \le -c^-(x). \quad (5.7d)$$

In these inequalities $A \equiv \nabla c(x)$. Also, Ω is a diagonal matrix such that $-\Omega A p$ is an element of the subdifferential of $c^-(x + \alpha p)$ at $\alpha = 0$. The diagonal entries of Ω take values in $[0, 1]$, are zero whenever $c_j(x) > 0$ and take the value one whenever $c_j(x) < 0$. Finally, $\omega^T A p$ represents an element of $\partial \varphi(0)$, the subdifferential of $\varphi(\alpha) \equiv \|c^-(x + \alpha p)\|_l$ at 0. The elements of ω are given by

$$\omega_j = (\Omega)_{jj} \left(\frac{c_j^-}{\|c^-\|_l} \right)^{l-1},$$

and have the property that $\omega^T c(x) = -\|c^-(x)\|_l$.

Consider now the case when ϕ has been defined from (5.1). From our assumption that $\lambda_k \ge 0$ and (2.4b),

$$\lambda_k^T \Omega_k(A_k p_k + c_k) \ge 0$$

for all k. It follows from this inequality and the relationships given in (5.7) that

$$\phi_k'(0) = g_k^T p_k + \xi_k^T c^-(x_k) - \lambda_k^T \Omega_k A_k p_k - \rho_k \omega_k^T A_k p_k \le \Delta_k.$$

We select ρ_k such that

$$\Delta_k \le -\tfrac{1}{2} p_k^T H_k p_k. \quad (5.8)$$

This rule is analogous to the ones used in Byrd and Nocedal [BN88], and Burke and Han [BH89].

The first step is to establish that such a value of ρ exists. From (3.13) and (5.3) we have

$$\Delta_k \le -(\tfrac{1}{2} + \beta_1)p_k^T H_k p_k + \beta_2 \|c_k^-\| - (\xi_k - \lambda_k)^T c_k^- - \rho\|c_k^-\|. \quad (5.9)$$

If we now use (2.6), property **MC1** and Lemma 2.1 to bound the multiplier term

$$(\xi_k - \lambda_k)^T c_k^- \le \|\mu_k - 2\lambda_k\|\|c_k^-\| \le 3\beta_\mu\|c_k^-\|,$$

we obtain in (5.9)

$$\Delta_k \le -(\tfrac{1}{2} + \beta_1)p_k^T H_k p_k + (\beta_2 + 3\beta_\mu - \rho)\|c_k^-\|.$$

Defining $\rho_u \equiv \beta_2 + 3\beta_\mu$, for any value $\rho \ge \rho_u$ condition (5.8) is satisfied for any k. This result also shows that the value of ρ will remain bounded in the algorithm.

Theorem 5.1. *The algorithm modified to use the merit function (5.1) converges globally.*

Proof. Given the bound in Lemma 3.8, it suffices to show that $\|p_k\| \to 0$.

As ρ cannot grow without bound, any strategy for increasing ρ by a finite quantity whenever it is required to increase ρ implies that there exists an iteration value K such that $\rho_k = \rho_K$ for all $k \geq K$. We consider only iterations of this form. For $k \geq K$, from (5.4), (5.8) and condition **MC2**,

$$\phi(\alpha_k) - \phi(\alpha_{k-1}) \leq \alpha_k \eta \Delta_k \leq -\eta \beta_{svH} \alpha_k \|p_k\|^2.$$

From the boundedness of ϕ (assumption **A3**), it follows that

$$\alpha_k \|p_k\|^2 \to 0. \tag{5.10}$$

If $\|p_k\| \to 0$, convergence follows from Lemma 3.8. Otherwise, if for a subsequence $\|p_k\| > \epsilon$, from (5.10) we must have $\alpha_k \to 0$ along the subsequence, and from the termination conditions for the linesearch (5.4), (5.5) and (5.6), $\bar{\alpha}_k \to 0$, as the step required to satisfy condition (2.16) is uniformly bounded away from zero (see (3.56) and (3.57)). Finally, from (5.6) we must also have $\hat{\alpha}_k \to 0$.

In the following relationships we drop the subscript k corresponding to the iteration number, and we denote by a tilde the value of functions evaluated at $x + \bar{\alpha}p$ (i.e.: $\tilde{c} \equiv c(x + \bar{\alpha}p)$).

From the definition of the merit function (5.1),

$$\begin{aligned}
\phi(\bar{\alpha}) - \phi(0) &= \bar{\alpha}g^T p + \lambda^T(\tilde{c}^- - c^-) + \bar{\alpha}\,\xi^T\tilde{c}^- - \bar{\alpha}\rho\|c^-\| \\
&\quad + (\tilde{F} - F - \bar{\alpha}g^T p) + \rho(\|\tilde{c}^-\| - (1 - \bar{\alpha})\|c^-\|).
\end{aligned}$$

For the last term, from (5.7a) and (5.7b), it follows that

$$\|\tilde{c}^-\| - (1 - \bar{\alpha})\|c^-\| \leq \|\tilde{c} - c - \bar{\alpha}Ap\|,$$

and

$$\begin{aligned}
\phi(\bar{\alpha}) - \phi(0) &\leq \bar{\alpha}g^T p + \lambda^T(\tilde{c}^- - c^-) + \bar{\alpha}\,\xi^T\tilde{c}^- - \bar{\alpha}\rho\|c^-\| \\
&\quad + (\tilde{F} - F - \bar{\alpha}g^T p) + \rho\|\tilde{c} - c - \bar{\alpha}Ap\|.
\end{aligned}$$

If we use again (5.7a) and (5.7b) on the terms associated with the multiplier estimates (given that by assumption $\lambda + \bar{\alpha}\xi \geq 0$), and the Taylor series expansions for F and c, we obtain

$$\begin{aligned}
\phi(\bar{\alpha}) - \phi(0) &\leq \bar{\alpha}g^T p + \sum_j (\lambda_j + \bar{\alpha}\xi_j)|\tilde{c}_j - c_j - \bar{\alpha}\,a_j^T p| + (1 - \bar{\alpha})\lambda^T c^- \\
&\quad - \lambda^T c^- + \bar{\alpha}(1 - \bar{\alpha})\xi^T c^- - \bar{\alpha}\rho\|c^-\| + O(\|\bar{\alpha}p\|^2).
\end{aligned}$$

After simplifying this expression we have

$$\phi(\bar{\alpha}) - \phi(0) \leq \bar{\alpha}\Big(g^T p + (\xi - \lambda)^T c^- - \rho\|c^-\|\Big) + \bar{\alpha}^2 \|c^-\|\,\|\xi\| + O(\|\bar{\alpha}p\|^2). \tag{5.11}$$

Replacing this bound in (5.6) implies

$$0 < (1-\sigma)\bar{\alpha}\Delta + \bar{\alpha}^2\|c^-\|\,\|\xi\| + O(\|\bar{\alpha}p\|^2). \tag{5.12}$$

Since from (5.8) and condition **HC2**, $\Delta \le -\beta_{svH}\|p\|^2$, and we have assumed that $\|p\| > \epsilon$, it follows by taking limits along the subsequence that

$$0 \le -(1-\sigma)\beta_{lvH}\epsilon^2.$$

However, this is not possible, implying $\|p_k\| \to 0$ for the whole sequence. ∎

Consider now the second merit function (5.2). The subgradient along the search direction at (x_k, λ_k) is given by

$$\phi'_k(0) = g_k^T p_k + \xi_k^T c^-(x_k) - \lambda_k^T \Omega_k A_k p_k - \rho_k c^-(x_k)^T A_k p_k \le \Delta_k,$$

where

$$\Delta_k \equiv g_k^T p_k + (\xi_k - \lambda_k)^T c^-(x_k) - \rho_k\|c^-(x_k)\|^2.$$

Note that $\lambda_k \ge 0$ implies

$$(\Omega_k \lambda_k + \rho_k c_k^-)^T (A_k p_k + c_k) \ge 0.$$

In this case it is not immediately evident that ρ_k remains bounded. The convergence proof we give is similar to the one introduced in Section 3. The definition of ρ given in that section will be preserved, except $c - s$ is replaced by c^-.

Theorem 5.2. *The algorithm modified to use the merit function (5.2) converges globally.*

Proof. Again, from Lemma 3.8 it is enough to show that $\|p_k\| \to 0$.

First assume that ρ is bounded. The argument used is similar to the one in Theorem 5.1. From (5.4), (5.8), condition **MC2** and the boundedness of ϕ, (5.10) must hold also for this case.

If $\|p_k\| \to 0$, convergence follows from Lemma 3.8. Otherwise, if for a subsequence $\|p_k\| > \epsilon$, from (5.10) we must have $\alpha_k \to 0$, and from condition (5.6) and the boundedness of the step to satisfy (2.16), $\hat{\alpha}_k \to 0$.

From (5.2), (5.7a) and (5.7b), we also have (we again drop the index k in the following relationships, and use a tilde to indicate values at $x + \bar{\alpha}p$)

$$\begin{aligned}
\phi(\bar{\alpha}) - \phi(0) \le\ & \bar{\alpha}g^T p + \lambda^T(\tilde{c}^- - c^-) + \bar{\alpha}\,\xi^T\tilde{c}^- - \rho(\bar{\alpha} - \tfrac{1}{2}\bar{\alpha}^2)\|c^-\|^2 \\
& + \rho\|\tilde{c} - c - \bar{\alpha}Ap\|\left(\tfrac{1}{2}\|\tilde{c} - c - \bar{\alpha}Ap\| + \|(c + \bar{\alpha}Ap)^-\|\right) \\
& + (\tilde{F} - F - \bar{\alpha}g^T p),
\end{aligned}$$

and again using (5.7a) and (5.7b) on the terms associated with the multiplier estimates, we obtain

$$\begin{aligned}
\phi(\bar{\alpha}) - \phi(0) \le\ & \bar{\alpha}\left(g^T p + (\xi - \lambda)^T c^- - \rho\|c^-\|^2\right) \\
& + \bar{\alpha}^2\|c^-\|\left(\|\xi\| + \tfrac{1}{2}\rho\|c^-\|\right) + O(\|\bar{\alpha}p\|^2). \tag{5.13}
\end{aligned}$$

Replacing this bound in (5.6) implies

$$0 < (1 - \sigma)\bar{\alpha}\Delta + \bar{\alpha}^2\|c^-\| \left(\|\xi\| + \tfrac{1}{2}\rho\|c^-\|\right) + O(\|\bar{\alpha}p\|^2). \tag{5.14}$$

Since from (5.8) and condition **HC2**, $\Delta \le -\beta_{svH}\|p\|^2$, and we have assumed that $\|p\| > \epsilon$ and ρ is bounded, it follows by taking limits along the subsequence that

$$0 \le -(1 - \sigma)\beta_{lvH}\epsilon^2.$$

However, this is not possible, which implies $\|p_k\| \to 0$ for the whole sequence.

Assume now that ρ_k grows without bound. In this case we have that for all iterations where the value of the penalty parameter is increased

$$\rho_{k_l}\|c_{k_l}^-\| \le K_1 \qquad \text{and} \qquad \rho_{k_l}\|p_{k_l}\|^2 \le K_2.$$

The proof of this result is basically that of Lemma 3.9. From these bounds it is possible to show that we must also have

$$\rho_k\|p_k\|^2 \le K$$

for all k (the proof is similar to the one for Lemma 3.10), implying $p_k \to 0$ and the convergence of the algorithm. ∎

6. Numerical Results

In this section we present numerical results obtained from an implementation of our algorithm. As a first step we have modified the code NPSOL. We have called the modified routine INPSOL. Apart from the definition of the search direction all other aspects of INPSOL are identical to those of NPSOL. A detailed description of NPSOL is given in Gill *et al.* [GMSW86a]. It should be noted that NPSOL does not incorporate linear constraints into the merit function. An initial point is obtained that is feasible with respect to the linear constraints and thereafter feasibility is retained (by incorporating the linear constraints in the QP subproblem). On many practical problems the feasible region with respect to the linear constraints is compact. On such problems this approach ensures assumption **A2** is satisfied, and assumption **A1** is implied by **A3**.

The purpose of the testing reported is to demonstrate that the efficiency and robustness of the modified algorithm are comparable to those of NPSOL. Naturally, we can only test the hypothesis on the domain of problems NPSOL is designed to solve, namely problems having a small number of variables and constraints, although on these problems the opportunities for improvement are limited, as we discuss later. What this implementation really tests is whether the introduction of flexibility in the determination of the search direction has a significant cost. The parameter β_c was set to infinity to avoid differences with NPSOL arising due entirely to the linesearch.

The search direction

The algorithm described in Section 2 allows for considerable flexibility of design. We describe here the specific choices made in our implementation. The search direction p_k is computed according to the following steps. (The subscript k is dropped from now on.)

- An initial feasible point for each QP subproblem, p_0, is obtained following the same procedure as NPSOL. No special effort was made to satisfy conditions (2.18) since on the problems tested no failure was detected that could be attributed to the size of $\|p_0\|$.

- The active-set method used in NPSOL was terminated at $\tilde{p}$, the *first* stationary point. The multipliers π at $\tilde{p}$ are then computed. Define $\hat{\pi}$ as $\hat{\pi}_j \equiv \pi_j \|a_j\|$.

- Let ϵ_M denote machine precision. If

$$\forall j, \quad \hat{\pi}_j \geq -\sqrt{\epsilon_M}, \tag{6.1}$$

then $\tilde{p}$ is taken as the search direction.

- If (6.1) does not hold a step that moves off a subset of the active constraints is computed. To identify the set of active constraints to be deleted, define $\pi_{\min} \equiv \min_j \hat{\pi}_j$, and introduce a vector e_I as

$$(e_I)_j \equiv \begin{cases} \|a_j\| & \text{if } \hat{\pi}_j \leq 10^{-3}\pi_{\min}, \\ 0 & \text{otherwise.} \end{cases} \tag{6.2}$$

- There is also a limit of 50 on the maximum number of constraints to be deleted. If (6.2) is satisfied by more than 50 active constraints, only the ones having the smallest multipliers are deleted. For most problems this limit has no effect, since the total number of constraints is less than 50.

- The direction d that moves off the selected constraints is obtained as the least-length solution of the system $Au = e_I$; that is, we define

$$d = Y(AY)^{-1}e_I,$$

where Y denotes a basis for the range-space of A^T.

- We obtain the search direction p from (2.21), as

$$p \equiv \begin{cases} \tilde{p} + \tilde{\gamma}d & \text{if } \|\tilde{p}\| < \beta_{slp}\|\tilde{p} + \tilde{\gamma}d\|, \\ \tilde{p} & \text{otherwise,} \end{cases}$$

where $\tilde{\gamma}$ was defined as in (2.24) with $\gamma_M = 10^{10}$ and $\beta_{slp} = 100$ (with this value the step $\tilde{p} + \tilde{\gamma}d$ is accepted in nearly all cases).

- Finally, the multiplier estimate used to define the linesearch is taken to be π if $p = \tilde{p}$. Otherwise, it is taken to be the least-squares estimate μ_L obtained from

$$AA^T\mu_L = Ag.$$

Test problems

The two algorithms, NPSOL and INPSOL, have been compared by solving a collection of 114 problems from the literature. The problems have been obtained from the following sources:

- Problem 1 is the example problem distributed with NPSOL; its description can be found in [GMSW86a]. Problems 3 and 4 are slight reformulations of the same problem, where the bounds $-1 \leq x_3 \leq 1$ have been replaced by the constraint $x_3^2 \leq 1$. Problem 3 uses the starting point

$$(\tfrac{1}{3}, \tfrac{2}{3}, \tfrac{11}{10}, \tfrac{2}{3}, \tfrac{1}{3}, \tfrac{1}{3}, \tfrac{2}{3}, -\tfrac{1}{3}, -\tfrac{1}{3}).$$

- Descriptions for problems 6 and 12–15 can be found in [MS82]. The version of problem 6 considered is the one corresponding to a value $T = 10$. Problems 12 and 13 start from point (d) for Wright No. 4 as indicated in the reference, while problems 14 and 15 start from points (a) and (b) for Wright No. 9, respectively.

- A description of the SQUARE ROOT problems (17–20) and of EXP6 (9) can be found in Fraley [Fra88].

- Problems 21–30 were obtained from Boggs and Tolle [BT84].

- All problems having names starting with "HS" are from Hock and Schittkowski [HS81].

- Problems 85–95 can be found in Dembo [Dem76].

All the above problems have been used in the past to test NPSOL. It should be noted that the problems in this group are small; the average number of variables is 10, and the average number of constraints is 6. Nevertheless, many of these problems are considered hard to solve. Moreover, for some of these problems the assumptions made to establish the convergence results fail to hold; for example, in some cases the Jacobian of the active NP constraints at x^* is singular, or no feasible points exist for some QP subproblems. In problem 42 no feasible point exists for NP.

The algorithms have also been tested on another group of problems.

- The structural optimization problems 99–114 are described in Ringertz [Rin88]. The letters "I" and "E" in the problem name indicate if the formulation used included explicitly the displacement variables ("E") or eliminated them in advance. Also, the following number (10, 25, 36 or 63) denotes the number of bars in the truss considered. Finally, whenever a number is included at the end of the name (006, 040 or 060), the initial point taken has been modified to be $x_j = 6$, 40 or 60 respectively.

These problems have been introduced due to the atypical behavior of quasi-Newton SQP algorithms on them. For this group, the ratio of QP to nonlinear iterations is large when compared to the size of the problem; on the first test set (problems 1–98) the average ratio for NPSOL is 2 QP iterations per nonlinear iteration, while on problems 99–114 the average ratio is 30.

The normal behavior of NPSOL on the first set of test problems is to require a relatively large number of QP iterations in the first few nonlinear iterations. Typically, the number of QP iterations declines exponentially until near a KKT point, when only one iteration is required. The STRUC problems depart from this "standard" behavior, in the sense that the number of QP iterations declines much more gradually. (Although only one QP iteration is required in the end, most nonlinear iterations require more.) This offers the possibility of observing the reductions that can be achieved by using the early-termination criterion, with limited distortion from the asymptotic behavior of NPSOL.

Finally, the problems in this second group are larger than the ones presented above; the average number of variables is now 55, and the average number of constraints is 100. For all the reasons mentioned, this set of problems provides a better environment in which to test the ability of the proposed early-termination criterion to reduce the number of QP iterations.

Computing environment

Version 4.02 of NPSOL was used in these comparisons. For this test set, all parameters used in the code have been fixed at their default values (see [GMSW86a]). No attempt was made to improve the results by selecting a different set of parameters. It would be difficult to compare the relative effort to adjust input parameters for the two algorithms. The runs were performed as batch jobs on a DEC VAXstation II with 5 Mb main memory. The operating system was VAX/VMS version 4.5, and the compiler used was VAX FORTRAN version 4.6 with default options.

Results

The results obtained from running both algorithms on the test set are presented in Table 2.

The parameters chosen to characterize the relative performance of both algorithms have been: the number of outer (nonlinear) iterations for each problem; the number of calls to the routine computing the values of the objective function, the constraint functions and their derivatives (function evaluations); the total number of inner (QP) iterations for the problem (this includes the number of iterations necessary to compute a feasible point); and the running (CPU) time needed to solve the problem. The results corresponding to both algorithms are given as a single entry in the tables, with the figures separated by a "/" symbol, in the form

<div align="center">NPSOL result/INPSOL result.</div>

Given that most of the problems are not convex, the algorithms may converge to different KKT points. Three such events occurred. Another possible outcome is failure—that is, the algorithm terminates without finding a solution, because the iteration limit has been exceeded, because no significant progress can be made at the current point with respect to the merit function, or because the objective or constraint functions need to be evaluated at a point for which they are not defined in the code. Such failures are indicated by "—".

For the set of 114 problems, NPSOL was able to find a KKT point in 107 cases, while INPSOL was able to solve 105 problems. We should emphasize that only the default value of the input parameters were used. Undoubtedly adjustment of the input parameters on the

problems that failed would have led to more successes. The figures illustrate the reliability of INPSOL.

Table 1 presents a summary of the results for the four quantities monitored in Table 2. The average values have been computed as the geometric means for the ratios of the values for NPSOL and for INPSOL; that is, averages larger than one indicate that the corresponding value for NPSOL is larger than the value for INPSOL. Also, the averages exclude those problems where one of the algorithms failed. Separate entries have been provided for problems 1–98 (the smaller problems), and for problems 99–114 (the structural optimization problems).

TABLE 1
Average Behavior: NPSOL vs. INPSOL

	Problems		
	All	1–98	99–114
Nonlinear iterations	.988	.979	1.044
Function evaluations	.994	.999	.963
QP iterations	1.190	1.112	1.884
CPU time	1.043	1.022	1.200

We now comment briefly on the implications of these results.

- The early-termination rule seems to behave very well regarding the numbers of nonlinear iterations and function evaluations; even if we are now using a search direction of "worse quality" than in NPSOL, the numbers are very close for both algorithms.

- The number of QP iterations is reduced by 20% for the complete set. When judging this figure we must take into account that the problems are small, implying that the number of QP iterations required per nonlinear iteration is also small. (In fact, the average value for the test set is 5.6 QP iterations per nonlinear iteration.) The opportunity for improvement is correspondingly limited. Moreover, both codes use the active set at the solution of the previous QP subproblem as a prediction for the correct active set in the current subproblem, resulting in a small number of QP iterations close to a KKT point. As a result, significant savings achieved by incomplete solution of QP subproblems in the early iterations are masked by a large number of subproblems requiring only a few QP iterations. As an example, for problem 98 the largest number of QP iterations needed in any nonlinear iteration is reduced from 57 for NPSOL to 15 for INPSOL. This effect is much less clear when we look at total numbers of QP iterations (244 for NPSOL vs. 170 for INPSOL). Recall that it is necessary in any implementation to limit the number of iterations taken to solve the subproblem. This large reduction in the maximum number of iterations is encouraging. Moreover, it

indicates that INPSOL and NPSOL took quite different paths to obtain a solution on many of the problems. In the light of this fact the similarity of performance is quite remarkable. Finally, the early-termination rule still requires a feasible point, and the feasibility phase is the same as in NPSOL. When this phase accounts for most of the total number of iterations, as with the STRUC problems, the possibility of improvement is further diminished.

Nonetheless, it should be noted that for problems 99–114 the improvement obtained is significantly greater than 20%, as the mean ratio is now 1.88; in fact, when we look only at the larger problems, the relative performance of INPSOL improves markedly. This offers the promise that for even larger problems the results obtained may be substantially better than the values shown above.

- The CPU time required by INPSOL is lower than the time for NPSOL, but by a factor that is much smaller than for the number of QP iterations. This is due not only to the fact that function evaluations can be expensive when compared to the effort to solve each QP subproblem, but also to some details in the implementation that have been chosen to affect the number of QP iterations, even at the expense of running time. For example, the multiplier estimate used for the linesearch (the least-squares multiplier) is expensive to compute when many constraints are deleted in the last step, as the factorization for the Jacobian of the active constraints must be updated. There are still options to be explored that might reduce the CPU time for the modified algorithm.

TABLE 2
Numerical Results

No.	Problem name	Nonlinear iterations	Function evaluations	QP iterations	CPU time (s)
1	NPSOL SAMPLE PROBLEM	12/13	16/18	45/34	3.69/3.61
2	SINGULAR	15/15	16/16	4/4	1.03/1.05
3	HEXAGON	15/16	21/23	32/29	4.41/4.41
4	HEXAGON (ALT. START)	11/11	16/14	35/26	3.56/3.26
5	LC7	7/9	9/11	13/16	.76/.95
6	ALAN MANNE'S PROBLEM	17/17	18/18	40/37	21.13/21.92
7	ROSEN-SUZUKI	8/8	11/11	9/9	.81/.81
8	QP PROBLEM	8/10	9/11	23/15	1.10/1.04
9	EXP6	33/53	35/57	38/57	1.96/3.08
10	STEINKE2	—*/5	—/6	—/14	—/.87
11	NORWAY	4/6†	5/7	34/13	1.23/.65
12	MHW4	10/10	18/15	14/12	1.31/1.25
13	MHW9	30/19†	56/28	42/24	3.71/2.31
14	MHW9 INEQUALITY 1	28/23	38/28	59/40	2.41/2.73
15	MHW9 INEQUALITY 2	41/14†	58/27	80/24	4.83/1.77
16	WOPLANT	25/29	29/33	44/35	6.85/7.17
17	SQUARE ROOT 1	—*/—*	—/—	—/—	—/—
18	SQUARE ROOT 2	23/23	36/36	0/0	5.01/5.32
19	SQUARE ROOT 3	6/6	9/9	7/7	.95/.94
20	SQUARE ROOT 4	—*/—*	—/—	—/—	—/—
21	BT1	11/11	19/19	11/11	.81/.83
22	BT2	9/9	14/14	9/9	.71/.70
23	BT3	2/2	5/5	2/2	.19/.19
24	BT4	12/12	18/18	13/13	.92/.92
25	BT5-HS63	6/6	9/9	8/8	.58/.58
26	BT6-HS77	15/15	21/21	16/16	1.52/1.54
27	BT7	31/31	56/56	32/32	3.36/3.43
28	BT8	17/17	19/19	17/17	1.25/1.44
29	BT9-HS39	13/13	16/16	14/14	.95/1.19
30	BT10	8/8	11/11	0/0	.48/.52
31	BT11-HS79	9/9	12/12	10/10	1.05/1.06
32	BT12	27/27	57/57	28/28	3.04/3.04
33	BT13	32/32	44/44	34/34	2.61/2.62
34	POWELL TRIANGLES	23/15	37/16	36/23	3.27/2.28
35	POWELL BADLY SCALED	12/12	15/15	13/13	.85/.85
36	POWELL WRIGGLE	34/32	69/55	60/40	2.77/2.39
37	POWELL-MARATOS	6/6	7/7	6/6	.44/.44
38	HS72	7/7	8/8	8/8	.69/.67
39	HS73 (CATTLE FEED)	4/4	5/5	4/4	.38/.36
40	HS107	11/11	18/18	27/18	2.77/2.56
41	MUKAI-POLAK	10/10	16/16	13/13	1.08/1.11
42	INFEASIBLE SUBPROBLEM	—*/—*	—/—	—/—	—/—
43	HS26	47/47	64/64	48/48	3.39/3.41
44	HS32	2/4	3/5	3/5	.25/.38
45	HS46	55/55	58/58	56/56	5.26/4.98
46	HS51	2/2	5/5	2/2	.18/.14
47	HS52	2/2	5/5	2/2	.19/.16
48	HS53	2/2	5/5	2/2	.19/.16
49	PENALTY1 A	16/16	18/19	77/41	20.01/16.49
50	PENALTY1 B	6/7	14/19	67/32	14.77/11.77
51	PENALTY1 C	29/15	85/40	152/65	24.35/11.65
52	HS13	22/19	23/20	13/10	1.29/1.22
53	HS64	29/43	39/62	47/60	2.34/3.33
54	HS65	8/9	10/11	16/16	.70/.78
55	HS70	36/—*	39/—	39/--	3.33/—
56	HS71	5/7	6/9	9/9	.53/.67
57	HS74	10/26	15/48	14/28	1.17/2.68

* Failed to solve the problem.
† Converged to a different minimizer.

TABLE 2 (CONT.)
Numerical results

No.	Problem name	Nonlinear iterations	Function evaluations	QP iterations	CPU time (s)
58	HS75	6/8	10/11	7/9	.72/.90
59	HS78	10/10	14/14	11/11	1.15/1.15
60	HS80	8/8	10/10	8/8	.92/.92
61	HS81	14/14	20/20	15/15	1.57/1.60
62	HS84	—*/4	—/5	—/9	—/.51
63	HS85	17/14	18/15	33/20	4.00/3.12
64	HS86 (COLVILLE 1)	6/7	8/8	11/11	.62/.64
65	HS87 (COLVILLE 6)	11/8	18/9	18/14	1.63/1.23
66	HS93	12/12	15/15	14/14	1.36/1.38
67	HS95	1/1	2/2	1/1	.15/.15
68	HS96	1/1	2/2	1/1	.17/.15
69	HS97	3/3	6/6	3/3	.40/.41
70	HS98	3/3	6/6	8/8	.43/.44
71	HS99	23/—*	44/—	74/—	3.99/—
72	HS100	14/14	29/29	18/18	2.07/2.02
73	HS104	18/18	20/20	23/23	3.36/3.37
74	HS105	43/—*	61/—	97/—	27.14/—
75	HS108 (HEXAGON)	24/32	45/49	57/87	6.78/9.36
76	HS109	11/10	13/11	25/29	3.23/3.26
77	HS110	6/6	9/9	24/15	.78/.69
78	HS111	41/49	64/75	44/52	8.08/9.05
79	HS112 (CHEMICAL EQ.)	19/—*	39/—	54/—	2.78/—
80	HS113	14/16	19/23	38/36	3.12/3.41
81	HS114	18/16	19/24	36/33	3.81/3.60
82	HS117 (COLVILLE 2)	17/18	21/27	96/39	6.75/5.34
83	HS118 (LC PROBLEM)	4/4	6/6	20/20	1.35/1.40
84	HS119 (COLVILLE 7)	12/17	16/19	41/47	4.25/5.60
85	DEMBO 1B	281/—*	437/—	296/—	75.46/—
86	DEMBO 2-HS83	4/4	6/6	4/4	.54/.54
87	DEMBO 3	9/8	11/9	37/20	2.01/1.78
88	DEMBO 4A	19/19	23/23	24/24	3.53/3.31
89	DEMBO 4C	13/13	15/15	20/23	3.10/3.20
90	DEMBO 5-HS106	17/18	21/24	30/31	2.90/3.04
91	DEMBO 6-HS116	36/43	96/69	144/248	21.84/29.65
92	DEMBO 7	19/12	24/15	126/68	15.54/9.82
93	DEMBO 8A	33/42	85/118	105/99	7.52/9.17
94	DEMBO 8B	29/29	69/71	88/73	6.51/6.45
95	DEMBO 8C	25/27	60/68	89/65	6.19/6.06
96	OPF	18/17	19/18	53/51	468.12/456.10
97	GBD EQUILIBRIUM MOD.	5/6	6/7	37/26	6.22/6.10
98	WEAPON ASSIGNMENT	96/73	98/76	244/170	120.78/114.93
99	STRUCI10KON	18/17	34/30	65/42	13.67/11.73
100	STRUCE10KON	26/29	49/67	87/84	17.68/20.75
101	STRUCI10VAN	23/19	41/34	54/51	16.30/13.85
102	STRUCE10VAN	—*/24	—/43	—/91	—/19.44
103	STRUCI25006	42/37	68/62	147/85	92.44/80.99
104	STRUCE25006	20/28	32/36	178/95	357.83/260.79
105	STRUCI25DAT	11/12	19/21	24/22	24.75/27.11
106	STRUCE25DAT	52/21	106/37	687/65	647.13/191.44
107	STRUCI36DAT	23/20	38/34	59/46	120.79/108.02
108	STRUCE36DAT	29/30	53/62	87/90	971.16/1021.87
109	STRUCI63040	117/112	211/202	6116/3091	8182.13/7159.03
110	STRUCE63040	375/—*	794/—	3545/—	77286.64/—
111	STRUCI63060	—*/98	—/244	—/3899	—/8281.02
112	STRUCE63060	63/115	150/316	6675/3407	25090.15/33228.42
113	STRUCI63DAT	246/136	354/412	9045/2060	12591.61/11424.54
114	STRUCE63DAT	52/72	86/145	8049/2858	41793.84/22740.66

* Failed to solve the problem.
† Converged to a different minimizer.

7. Acknowledgement

We are grateful to the referees for their effort at refereeing a long and difficult paper. Their care and attention to detail resulted in a substantial improvement over the first version of this paper.

8. References

[Big72]   M.C. Biggs (1972), Constrained minimization using recursive equality quadratic programming, in: F.A. Lootsma, Ed., *Numerical Methods for Nonlinear Optimization* (Academic Press, London/New York).

[BT84]   P.T. Boggs and J.W. Tolle (1984), A family of descent functions for constrained optimization, *SIAM Journal on Numerical Analysis* 21 1146–1161.

[BTW82]   P.T. Boggs, J.W. Tolle and P. Wang (1982), On the local convergence of quasi-Newton methods for constrained optimization. *SIAM Journal on Control and Optimization* 20 161–171.

[BH89]   J.V. Burke and S-P. Han (1989), A robust sequential quadratic programming algorithm, *Mathematical Programming* 43 277–303.

[BN88]   R.H. Byrd and J. Nocedal (1988), An analysis of reduced Hessian methods for constrained optimization, Report CU-CS-398-88, Department of Computer Science, University of Colorado.

[CM87]   P.H. Calamai and J.J. Moré (1987), Projected gradient methods for linearly constrained problems, *Mathematical Programming 39 93–116.*

[CDT85]   M.R. Celis, J.E. Dennis, Jr. and R.A. Tapia (1985), A trust region strategy for nonlinear equality constrained optimization, in: P.T. Boggs, R.H. Byrd and R.B. Schnabel, Eds., *Numerical Optimization 1984* (SIAM, Philadelphia).

[Dem76]   R.S. Dembo (1976), A set of geometric programming test problems and their solutions, *Mathematical Programming* 10 192–213.

[DT85]   R.S. Dembo and U. Tulowitzki (1985), Sequential truncated quadratic programming methods. in: P.T. Boggs, R.H. Byrd and R.B. Schnabel, Eds., *Numerical Optimization 1984* (SIAM. Philadelphia).

[Eld91]   S.K. Eldersveld (1991), Large-scale sequential quadratic programming algorithms. Ph. D. Thesis, Stanford University.

[Fle70]   R. Fletcher (1970), A class of methods for nonlinear programming with termination and convergence properties, in: J. Abadie, Ed., *Integer and Nonlinear Programming* (North Holland, Amsterdam).

[Fle85]   R. Fletcher (1985), An ℓ_1 penalty method for nonlinear constraints, in: P.T. Boggs, R.H. Byrd and R.B. Schnabel, Eds., *Numerical Optimization 1984* (SIAM, Philadelphia).

[Fle87]   R. Fletcher (1987), *Practical Methods of Optimization* (John Wiley and Sons, Chichester/New York/Brisbane/Toronto/Singapore).

[Fra88]   C. Fraley (1988), Software performance on nonlinear least-squares problems, SOL Report 88-17, Department of Operations Research, Stanford University.

[GMSW86a]   P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright (1986), User's guide for NPSOL (Version 4.0): a FORTRAN package for nonlinear programming, Report SOL 86-2, Department of Operations Research, Stanford University.

[GMSW86b]   P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright (1986), Some theoretical properties of an augmented Lagrangian merit function, Report SOL 86-6R, Department of Operations Research, Stanford University.

[GMSW91]   P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright (1988), Inertia-controlling methods for quadratic programming, *SIAM Review* 33, 1–33.

[GMW81]   P.E. Gill, W. Murray and M.H. Wright (1981), *Practical Optimization* (Academic Press, London/New York).

[GO85]   J. Goodman (1985), Newton's method for constrained optimization, *Mathematical Programming* **33** 162–171.

[GO89]   C.B. Gurwitz and M.L. Overton (1989), Sequential quadratic programming methods based on approximating a projected Hessian matrix, *SIAM J. Sci. Stat. Comput.* **10** 631–653.

[Han76]   S.-P. Han (1976), Superlinearly convergent variable metric algorithms for general nonlinear programming problems, *Mathematical Programming* **11** 263–282.

[HS81]   W. Hock and K. Schittkowski (1981), Test examples for nonlinear programming, Lecture Notes in Economics and Mathematical Systems, Vol. 187 (Springer-Verlag, Berlin/Heidelberg/New York).

[MS84]   J.J. Moré and D.C. Sorensen (1984), Newton's method, in: G.H. Golub, Ed., *Studies in Numerical Analysis* (Mathematical Association of America) 29–82.

[Mu69]   W. Murray (1969), An algorithm for constrained minimization, in: R. Fletcher, Ed., *Optimization* (Academic Press, London/New York).

[MS82]   B.A. Murtagh and M.A. Saunders (1982), A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints, *Mathematical Programming Study* **16** 84–117.

[Po78]   M.J.D. Powell (1978), A fast algorithm for nonlinearly constrained c.... ..ns, in: O.L. Mangasarian, R.R. Meyer and S.M. Robinson, Eds., *Nonlinear Programming 3* (Academic Press, New York).

[Po87]   M.J.D. Powell (1987), Methods for nonlinear constraints in optimization calculations, Proceedings of the 1986 IMA/SIAM Meeting on "The State of the Art in Numerical Analysis", Clarendon Press, Oxford.

[PY86]   M.J.D. Powell and Y. Yuan (1986), A recursive quadratic programming algorithm that uses differentiable exact penalty functions, *Mathematical Programming* **35** 265–278.

[Pr89]   F.J. Prieto (1989), Sequential quadratic programming algorithms for optimization, Report SOL 89-7, Department of Operations Research, Stanford University.

[Rin88]   U.T. Ringertz (1988), A mathematical programming approach to structural optimization, Report No. 88-24, Dept. of Aeronautical Structures and Materials, The Royal Institute of Technology, Stockholm.

[Rob74]   S.M. Robinson (1974), Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear programming algorithms, *Mathematical Programming* **7** 1–16.

[Sch81]   K. Schittkowski (1981), The nonlinear programming method of Wilson, Han and Powell with an augmented Lagrangian line search function, *Numerische Mathematik* **38** 83–114.

[Wil63]   R.B. Wilson (1963), A simplicial algorithm for concave programming, Ph.D. Thesis, Harvard University.

[Wri76]   M.H. Wright (1976), Numerical methods for nonlinearly constrained optimization, Ph.D. Thesis, Stanford University.

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget. Paperwork Reduction Project (0704-0188), Washington, OC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE June 1993	3. REPORT TYPE AND DATES COVERED Technical Report

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
A Sequential Quadratic Programming Algorithm Using an Incomplete Solution of the Subproblem	Grant N00014-90-J-1242

6. AUTHOR(S)

Walter Murray and Francisco J. Prieto

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Department of Operations Research - SOL Stanford University Stanford, CA 94305-4022	1111MA

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
Office of Naval Research - Department of the Navy 800 N. Quincy Street Arlington, VA 22217	SOL 93-3

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
UNLIMITED	UL

13. ABSTRACT (Maximum 200 words)

We analyze sequential quadratic programming (SQP) methods to solve nonlinear constrained optimization problems that are more flexible in their definition than standard SQP methods. The type of flexibility introduced is motivated by the necessity to deviate from the standard approach when solving large problems. Specifically we no longer require a minimizer of the QP subproblem to be determined or particular Lagrange multiplier estimates to be used. Our main focus is on an SQP algorithm that uses a particular augmented Lagrangian merit function. New results are derived for this algorithm under weaker conditions than previously assumed; in particular, it is not assumed that the iterates lie on a compact set.

14. SUBJECT TERMS	15. NUMBER OF PAGES 53 pp.
quasi-Newton; nonlinear programming; sequential quadratic programming.	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED			SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102